Computer Simulation of a Plasma Accelerator

by

Velvin Richard Watson

May 1969

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of Office of Naval Research, Washington, D.C., Code 427.

SUIPR Report No. 313

Prepared under
Office of Naval Research Contract
Nonr-225(83), NR 373 360
Jointly supported by the U. S. Army Signal Corps, the
U.S. Air Force, and the U. S. Navy
(Office of Naval Research)

INSTITUTE FOR PLASMA RESEARCH
STANFORD UNIVERSITY, STANFORD, CALIFORNIA

163

COMPUTER SIMULATION OF A PLASMA ACCELERATOR

by

Velvin Richard Watson

May 1969

Report SUIPR No.313
Prepared under
Nonr-225(83)

Institute for Plasma Research
Stanford Electronics Laboratories
Stanford University   Stanford, California

Abstract

A computer is used to simulate a collisionless plasma in an axisymmetric high specific impulse plasma accelerator with a constant applied electric field and with no applied magnetic field. The collisionless plasma model is shown to be applicable to plasma accelerators that operate with specific impulses in excess of 10^4 seconds and with particle number densities of the order of 10^{15} particles/cm^3 or less. In particular, the model is used to simulate an accelerator that has recently produced specific impulses of the order of 10^5 seconds with particle number densities of the order of 10^{15} particles/cm^3.

The computer simulation is accomplished by representing the many ions and electrons in the plasma by several thousand representative particles. These several thousand particles are assumed to constitute a good statistical sample of the actual ions and electrons, and the normalized distribution function of the plasma is approximated by the normalized distribution function of the representative particles. The motion of these representative particles is followed ~~in time~~, and the electromagnetic fields induced by these particle motions are calculated by integrating the charge and current densities obtained from the particle distributions.

A computer program has been developed to display simultaneously on a cathode ray tube the electric potential, the magnetic field, and the motion of the representative particles as they move through the accelerator under the influence of the fields. These displays, which have been recorded on motion picture film, illustrate the coupling between the particle motions and the fields. In particular, the dis-

plays illustrate how the plasma motion modifies the applied electric potential, how the currents induce large magnetic fields, how the ions receive their energy directly from the electric field, and how the ions are deflected toward the accelerator exhaust by the induced magnetic field.

The results of the computer simulation illustrate that two frequently used concepts for a nearly neutral highly conducting plasma should not be applied to this problem. These concepts are (1) that the $\rho_i\vec{E}$ force cancels the $\rho_e\vec{E}$ force for a neutral plasma, and (2) that plasmas with infinite conductivity are frozen to magnetic field lines. Although the plasma is nearly neutral, the $\rho_i\vec{E}$ force is not strictly cancelled by the $\rho_e\vec{E}$ force as in a highly collisional plasma because the electron and ion gases can easily slip through one another. For this case, the $\rho_e\vec{E}$ force on the electrons is balanced almost entirely by the $\vec{v}_e\times\vec{B}$ force rather than by collisions with the ions, and the $\rho_i\vec{E}$ force is the dominant force for adding energy to the ions just as in the electrostatic ion gun. Although the electrical conductivity in the simulation is effectively infinite because there are no particle-particle collisions included in the simulation, the fluid is not frozen to the magnetic field lines because of the large ion inertia forces.

Although the ions are accelerated by an electrostatic field just as in an ion gun, the current is not space-charge limited as in an ion gun because electrons are present to partially neutralize the plasma. The electrons are restricted from flowing directly to the anode by the self-induced magnetic field.

Since the ions receive their energy from an electrostatic field, they do not obtain the "anomalous velocities" (i.e., velocities greater

than they would receive by falling through the potential applied across the electrodes) that are reported for the MPD arcjets.

Performance predictions based on the computer simulation are given for the high specific impulse accelerators for applied voltages from 10^3 to 10^5 volts.

The results from the simulation suggest that the high specific impulse plasma accelerators should operate most efficiently with no applied magnetic field and with a mass flow rate proportional to the current. The specific impulse should increase in proportion to the current, and the voltage and thrust should increase with the square of the current.

ACKNOWLEDGMENTS

I am grateful to Professor Oscar Buneman for encouragement and guidance in this research.

The following people generously spent their time with me in informal discussions and gave me much valuable information:

William Carlson	Ames Research Center, NASA
Howard Stine	Ames Research Center, NASA
Dr. Thomas Stratton	Los Alamos Scientific Laboratory, University of California
T. D. Butler	Los Alamos Scientific Laboratory, University of California
Professor Dah Yu Cheng	University of Santa Clara
Professor Robert Jahn	Princeton University
Dr. Kenn Clark	Princeton University
Dr. Richard John	AVCO

I am also grateful to the computer staff of the Stanford Research Institute for their assistance in developing the computer program to display the computer simulations on a cathode ray tube.

I am indebted to the following people who helped in the preparation of the manuscript: Professors Daniel Bershader and I-Dee Chang, who read the manuscript and made a number of helpful suggestions; Miss Ann Morris, who conscientiously typed the manuscript; and my wife, Sandy, who spent many hours with me checking the manuscript.

The research in this dissertation was supported by the Ames Research Center of NASA and by the Office of Naval Research under contract #Nonr 225(83).

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF SYMBOLS

A	= atomic weight
$\vec{A}$	= magnetic potential vector, Weber/m
$\vec{B}$	= magnetic field, tesla
B_θ	= azimuthal magnetic field, tesla
C_1,C_2,C_3	= proportionality constants of equations (1), (3), and (4)
e	= charge of a singly ionized ion, coulombs
$\vec{E}$	= electric field, V/m
I	= current, A
$\vec{J}$	= current density, A/m^2
$\vec{J}_e$	= electron current density, A/m^2
$\vec{J}_i$	= ion current density, A/m^2
$\vec{J}_z$	= axial current density, A/m^2
L	= characteristic dimension of the accelerator, m
m_e	= electron mass, kg
m_i	= ion mass, kg
$\dot{m}$	= mass flow rate, kg/s
n_e	= electron number density, electrons/m^3
r	= radial position within the accelerator, m
r_g	= gyro radius, m
v_e	= electron velocity, m/s
v_i	= ion velocity, m/s
Z	= number of charges on each ion
z	= axial position within the accelerator, m
Δt	= incremental time step for the computer simulation, s
ϵ	= electric permittivity, farads/m

λ_e = electron mean free path between collisions, m

μ = magnetic permeability, N/A^2

ρ = density, kg/m^3

ρ_{e-} = electron charge density, coulombs/m^3

ρ_i = ion charge density, coulombs/m^3

$\hat{\rho}$ = relative charge density = $\frac{\rho_i - \rho_e}{\rho_e}$

τ_e = mean time between electron collisions, s

Φ = potential applied across the electrodes, V

ω = electron gyrofrequency, radians/s

I. INTRODUCTION

Electric plasma accelerators are attractive for deep space propulsion because they offer a required combination of high specific impulse and high thrust density. The high specific impulse is required for efficient use of the propellant, and the high thrust density is required to keep mission times within reasonable bounds.

A graph produced by Clark and Jahn[1] that compares the specific impulses and thrust densities of electric accelerators with specific impulses and thrust densities of other propulsion devices is shown on figure 1. This graph has been modified to show the new state of the art for pulsed plasma guns as given by Cheng[2]. The graph illustrates that other propulsion devices can produce either high specific impulse or high thrust density, but not both.

The chemical rocket can produce high thrust density but not high specific impulse. It converts chemical energy to thermal energy and then converts thermal energy to directed energy with a nozzle. With this arrangement, the directed energy per unit mass of propellant cannot exceed the chemical energy per unit mass available for conversion. The chemical rocket can achieve a high thrust density by using a large mass flow of propellant, but the velocity and the specific impulse are limited as shown on figure 1. On the other hand, the ion engine can produce high specific impulse but cannot produce high thrust density. The flow of ions that are accelerated electrostatically by the high voltage acceleration grid is limited by the space charge of the ions between the ion source and the acceleration grid. Therefore, even though the ion engine can accelerate each ion to high velocity, it can not produce high thrust density. The nuclear rocket, which converts nuclear energy to thermal energy and then thermal energy to directed energy with a nozzle, can produce high thrust density but is limited in specific impulse by the properties of the material containing the heated propellant. Although a large amount of energy per unit of mass can be converted from nuclear energy to thermal energy, the propellant must be contained by the physical wall; so the thermal energy is limited, and consequently the specific impulse is limited to the value shown on figure 1. In a resistojet, an electrically conducting material is heated electrically and the propellant is in turn heated by the conducting material. The propellant temperature is always less than the temperature of the conducting wall, and consequently the specific impulse is limited to the values shown on figure 1.

The remaining devices shown on figure 1 are electric plasma accelerators. The conventional arcjet uses an electric arc to heat the gas and then converts the thermal energy to directed energy with a nozzle. These arcjets are limited in specific impulse by the properties of the material containing the heated propellant exactly the same as the nuclear rocket. The cross-field, MPD, and Cheng's accelerators can achieve much higher specific impulses because the electrical energy is not all converted to thermal energy -- rather much of the energy is converted straightway into directed energy, and, consequently, no nozzle is required to convert thermal energy into directed energy. The specific impulse achievable is then limited by the amount of electrical energy that can be converted straightway into directed energy without impinging excessive energy onto the material components of the accelerator. Some of the propellant impinges on the accelerator wall and the electrical current carried by the electrodes imparts heat to the electrodes; so the amount of energy that can be imparted to the gas will ultimately be limited even in the devices that convert a high fraction of the total electrical energy directly into directed energy. This limit has not yet been accurately determined. Nevertheless, it has been shown experimentally that the combination of specific impulse and thrust density already obtainable with the electric plasma accelerators is much greater than that obtainable by the other propulsion devices.

Plasma accelerators also have an application as a source for high energy plasma for controlled nuclear fusion devices. All the plasma containment devices designed thus far for continuous operation lose plasma at a fairly rapid rate, and in order to replace the lost plasma a source of high energy plasma is required. Plasma accelerators have the capability to produce a plasma with the energy required for fusion at mass flow rates orders of magnitude larger than the ion guns presently being used.

The development of the electric plasma accelerator has been primarily achieved by experimental trial and error techniques because a good, detailed theoretical analysis was not available. At the end of 1963, Ducati[3] found experimentally, to his surprise, that the specific impulse within an arcjet, such as the one shown in figure 2, could be increased from the region on figure 1 denoted as the conventional arcjet to the region denoted as the MPD arcjet by simply reducing the pressure in the accelerator from near atmospheric pressure to pressures under 0.1 atmosphere, and by increasing the current from hundreds of amperes to thousands of amperes. In 1968, Cheng obtained the further increased performance shown on figure 1 from an experiment in which he maintained a low pressure and further increased the current from thousands of amperes to the order of one-hundred thousand amperes within a coaxial configuration as shown on figure 3. Clark and Jahn[4] are experimenting with an electric accelerator with the configuration shown on figure 4 at currents in excess of 100,000 amperes, but they have not yet measured the specific impulse and thrust density within their accelerator. Although Cheng's and Clark and Jahn's accelerators are pulsed, the characteristics of the accelerators are nearly steady over most of the pulse, and Clark and Jahn[4] point out that the performance is a good simulation of steady state accelerators.

Disguised by the seemingly continuous, smooth transition of the experimental results from the conventional arcjet regime to the MPD arcjet regime and onto the regime of Cheng's data is the fact that the relative importance of the various accelerating mechanisms differ appreciably in the various regimes. Theoretical models have recently been developed to describe the acceleration mechanisms for the conventional arcjet and for the MPD arcjet. The purposes of this dissertation are (1) to present a theoretical investigation of the acceleration mechanisms for an electric accelerator in the high performance regime of Cheng's data, (2) to use the results of this investigation to argue that the models recently developed to approximate the acceleration mechanisms for the MPD arcjet regime cannot be extended into the high performance regime, (3) to suggest a more appropriate theoretical model to be used for this regime, and (4) to present results of a computer simulation based on this model.

II. ACCELERATING MECHANISMS WITHIN ELECTRIC PLASMA ACCELERATORS

The shift in importance of the different accelerating mechanisms can be related to the importance of particle collisions within the accelerator. When the particle free paths are sufficiently small, as in the conventional arcjets that operate at near atmospheric pressure, the electrical energy given to the charged particles is quickly randomized by collisions. No appreciable directed energy is imparted to the gas by the electromagnetic field -- instead the thermal energy is converted to directed energy with a nozzle.

As the charged particle path lengths between encounters increase, the charged particles are deflected between collisions by the magnetic field and receive a component of velocity in the $\vec{J}x\vec{B}$ direction. If collisions are sufficiently frequent, most of the electrical energy is given to the electrons, and the electrons transfer their directed energy to the ions by collisions. However, if collisions are sufficiently rare, the electrons cannot transfer their energy to the ions by close collisions and the ions receive most of their energy directly from the collective electromagnetic fields.

To determine whether or not the electrons can transfer their directed energy to the ions by close collisions, one must compare the electron path lengths required to transfer energy to the ions with the dimensions of the accelerators. (This path length is much longer than a "free path" because the energy exchange in an elastic collision between a light particle and a heavy particle is small compared to the kinetic energy of the particles). Accelerator dimensions are usually of the order of centimeters. The MPD arcjet, Cheng's accelerator, and Clark and Jahn's accelerator operate with ion number densities of the order of 10^{15} ions/cm^3 throughout most of the accelerator. For this density, the minimum electron path length required to transfer to hydrogen ions energies corresponding to specific impulses from 10^3 to 10^5 seconds have been estimated and are shown in figure 5. These minimum path lengths were calculated using the coulomb cross section and assuming that the average velocity of electrons is equal to the average ion exit velocity. Electron velocities are most likely higher than this minimum value; therefore, the corresponding path lengths are most likely longer than the minimum values shown in figure 5. The path length increases rapidly with increasing energies because the coulomb cross section decreases rapidly with increasing energy. Figure 5 shows that for an electron to transfer 10 ev to the ions a path length of the order of 1 cm is required, whereas a 10^6 cm path is required to transfer 10^4 ev.

The range of specific impulses depicted by Clark and Jahn for the MPD arcjet is also shown on figure 5, and it is seen that the corresponding path lengths are of the order of or smaller than the dimensions of a typical accelerator. Therefore, in such a device the electrons can transfer to the ions by collisions much of the energy directed in the $\vec{J}x\vec{B}$ direction. When, in addition, the number of electrons that pass through the accelerator is large compared to the number of ions that pass through the accelerator (electron current much greater than ion current) each ion can be given energy from many electrons. Thus, the ion can achieve an "anomalous velocity", i.e. a velocity greater than it would receive by falling freely through the available potential field.

On the other hand, in Cheng's experiment, the specific impulse is of the order of 10^5 secs and the electron path length shown on figure 5 for this specific impulse is orders of magnitude larger than the largest dimension of his accelerator. Therefore, either (1) many more electrons than ions flow so even though each electron gives only a small increment of energy to an ion sufficient electrons are present to give the relatively few ions their final energy; or (2) the ions receive most of their energy directly from the electomagnetic fields. If the former were correct, most of the electrons would go to the anode because, for steady operation of an electrically floating accelerator, the number of electrons leaving the accelerator must equal the number of ions leaving the accelerator. Since most of the energy of the electron could not be transferred to the ions, most of the energy would be transferred to the anode resulting in efficiencies orders of magnitude less than unity. Although efficiencies have not been accurately measured in Cheng's accelerator, order of magnitude measurements show that they are not very small compared to unity. Therefore, the latter must be correct -- most of the energy the ions receive must come directly from the electromagnetic fields (induced as well as applied) rather than from close collisions with electrons. (For steady fields, the ion must receive its energy from the electric field alone because a steady magnetic field can only deflect a charged particle rather than impart energy to it. Consequently, high specific impulse in a steady accelerator requires high voltage.)

Although the electron cannot transfer much energy to an ion during an electron-ion collision, the direction of motion of the electron is appreciably changed by the collision. Nevertheless, when the mean free path for the electron is much greater than the radius of gyration of the electron in the magnetic field, (ie, when the Hall parameter $\equiv \omega\tau_e = \frac{\lambda_e}{r_g}$ is much greater than one), the mean motion of the electrons is nearly the same as if they were collisionless. Between collisions the mean velocity of the electrons is equal to $\vec{E}x\vec{B}/B^2$. A collision merely shifts the center of gyration, but the velocity remains at $\vec{E}x\vec{B}/B^2$. The shift in the center of gyration is of the order of the gyro radius; so as long as the electron moves much further than a gyro radius between collisions, the mean motion of the electrons will be nearly the same as if the electrons were collisionless. In Cheng's experiments, the magnetic field was of the order of one tesla (10^4 gauss). The electron temperature is not known, but even if it were as low as 5 ev, the resulting Hall parameter would be greater than 100. Therefore, the motion of the electrons, as well as the ions, is little influenced by collisions.

Although collisions are not a primary mechanism for accelerating the ions, collisions are important for producing the plasma. The following more detailed examination of the processes that occur within the high specific impulse accelerator indicates how there may be sufficient collisions to produce a plasma but not for effective transfer of energy between electrons and ions.

The ion density throughout most of the accelerator is on the order of 10^{15} ions/cm^3, but the neutral particle density near the point of gas injection must be much greater. A speculation of the heavy particle density within Cheng's accelerator is given in figure 6. This distribution was deduced from the measured mass flow rate, the one-dimensional continuity equation, a measured ion exit velocity, and a speculated variation of axial velocity with axial position. The measured velocity at the exit is approximately 10^6 m/s whereas the velocity near the region of gas injection should be of the order of the thermal velocity of the gas: less than 10^4 m/s. The velocity curve shown in figure 6 is merely a speculation of how the velocity increases from the order of 10^4 m/s to 10^6 m/s; the large difference in magnitude from the inlet to the exit is the important feature rather than the shape of the curve. With a hydrogen flow rate of the order of 1 g/s and with a cross sectional area of approximately 10 cm^2, the particle densities obtained from this velocity curve are as shown on figure 6. The density near the inlet is two orders of magnitude higher than the density throughout the rest of the accelerator.

Most of the ionization should occur in the high density region. After a particle is ionized, it suddenly experiences the high accelerating force of the electromagnetic fields. (Note that the ionization energy and the thermal energy of the gas at the inlet are small compared to the kinetic energy of the ions at the accelerator exhaust). While the ion has low energy, it should continue to collide frequently, but as it receives more and more energy its collision cross section decreases rapidly and it moves into a low density region. By the time a hydrogen ion has received one-tenth of its energy and has moved into the low density region, its mean free path for collisions with other ions (shown in figure 7) is already several orders of magnitude larger than the size of the accelerator. Furthermore, in this low density region, there are insufficient electron-ion collisions to transfer an appreciable energy between electrons and ions. Therefore, the hydrogen ions entering the low density region should become effectively runaway particles, falling for all practical purposes freely through the electromagnetic fields. (Since they receive most of their energy in this region where they are essentially freely falling, it is important that theories for these accelerators model this "free fall region" correctly.)

III. THEORETICAL MODELING OF THE ELECTRIC PLASMA ACCELERATORS

The theoretical models used to describe the electric plasma accelerators must reflect the changes in the accelerating mechanisms described in the previous section.

A typical conventional arcjet is shown in figure 2. Most of the electrical energy is converted to thermal energy in the constricted part of the nozzle where the pressure and the electrical resistance are relatively high. For this constricted region, one can model the energy conversion and the gasdynamics with continuum fluid equations with the ohmic heating term, $\vec{J}\cdot\vec{E}$, included in the energy equation but with the electromagnetic force terms, $\rho\vec{E}$ and $\vec{J}x\vec{B}$, neglected in the momentum equation. Theoretical analyses for the constricted region using this model have been presented by Stine and Watson[5] and by Watson and Pegot[6]. There is relatively little electrical energy converted to thermal energy in the expanded part of the nozzle, and this region may be approximated using the continuum fluid equations with none of the electromagnetic terms.

A typical MPD arcjet is shown in figure 8. In the MPD arcjet regime, the charged particles are given an appreciable deflection between collisions by the magnetic field. The resulting force on the electron, expressed by the $\vec{J}_e x\vec{B}$ term is in the same direction as the resulting force on the ion, expressed by the $\vec{J}_i x\vec{B}$ term, and the total force, expressed by $\vec{J}x\vec{B}$, is not negligible compared to the pressure force as it was in the conventional arcjet. The forces on the charged particles due to the electric field, $\rho_e\vec{E}$ and $\rho_i\vec{E}$, are opposing, and because there are sufficient collisions such that the electrons and ions can transfer energy, the net force, $\rho\vec{E}$, is negligible compared to the $\vec{J}x\vec{B}$ force in this regime. Therefore the theoretical models for the MPD arcjet regime include the $\vec{J}x\vec{B}$ term in the continuum momentum equation but neglect the $\rho\vec{E}$ term.

A number of attempts have been made to obtain solutions for the properties of the MPD arcjet using this theoretical model. These attempts have been reviewed recently by Nerheim and Kelly[7] who concluded that "the thrust production mechanisms are poorly understood as indicated by the number of conflicting assumptions, approximations, and processes that have been proposed." Although the only forces considered for these attempts have been the pressure forces and the $\vec{J}x\vec{B}$ forces, an accurate solution of the pressure, current, and magnetic fields throughout the accelerator has not been obtained, and the various assumptions and approximations referred to by Nerheim and Kelly are those that are made concerning these fields. In some analyses the $\vec{J}x\vec{B}$ forces are considered as constraint forces and the propellant is assumed to be constrained to flow parallel to the magnetic field so that the magnetic field surfaces act as a nozzle. In other analyses the $\vec{J}x\vec{B}$ forces are divided into components and each component is treated separately. The axial component is called the "blowing force" and the radial component is called a "pumping force." When there is an applied magnetic field or if the properties are not axisymmetric, there is also an azimuthal component. There is considerable controversy regarding the relative importance of these various components of the $\vec{J}x\vec{B}$ force and regarding how they should be calculated. Nevertheless, all of the analyses for the MPD arcjet regime use this same model wherein the $\vec{J}x\vec{B}$ forces are included and the $\rho\vec{E}$ forces are neglected in the continuum momentum equation.

In the direct electromagnetic acceleration regime, the electrons cannot transfer an appreciable fraction of their energy to the ions, and the ions receive a major part of their energy from falling through the electric field. Therefore, the $\rho_i\vec{E}$ term becomes the dominant force term for imparting energy to the ions. Furthermore, throughout most of the accelerator, neither the mean free path of the ions, shown in figure 7 as a function of ion energy, nor the ion Larmor radius is small compared to the size of the typical accelerator; so the continuum equations are inappropriate. Therefore, the basic model used for the MPD arcjet regime which employs the continuum equations and neglects acceleration directly due to the electric field is not appropriate in the interesting regime of the direct electromagnetic accelerators.

The theoretical model used herein for the high specific impulse accelerators operating in the regime of direct electromagnetic acceleration gives prime consideration to the "free fall region" described in the previous chapter because the ions receive most of their energy in this region. The high density region near the point of gas injection is considered only as a source of plasma for the upstream edge of the "free fall region". The plasma in the "free fall region" is considered as collisionless and this collisionless plasma is simulated with a computer as described in the next chapter.

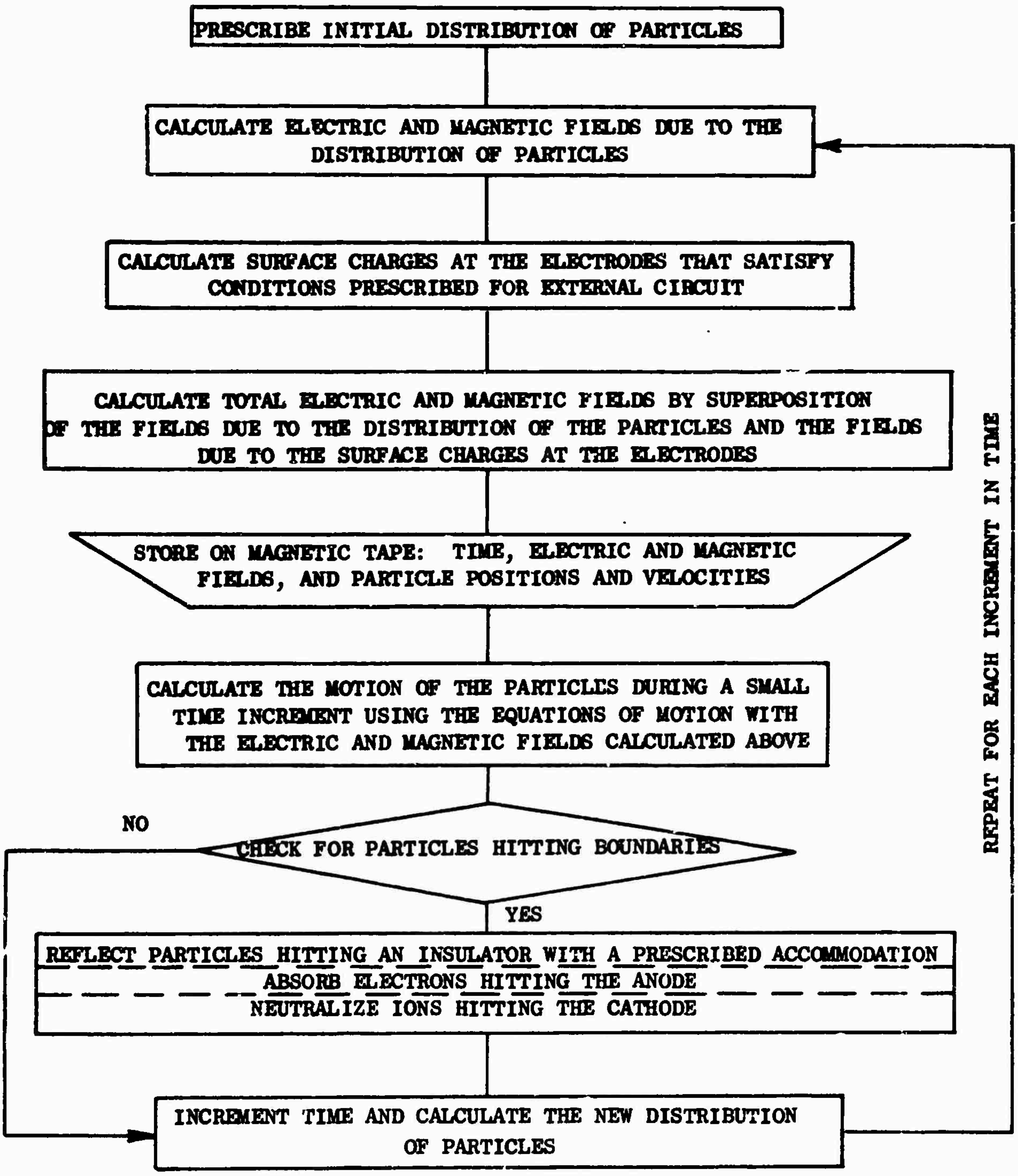
In 1964, in discussions with Professor R. Jahn of Princeton and Dr. R. John of AVCO, the author proposed that a regime of direct electromagnetic acceleration would exist and suggested a simple model to describe the qualitative features of axisymmetric accelerators in this regime. The model treats individual particle trajectories rather than the continuum equations, and the ions are assumed to receive their energy directly from the electric fields. At about the same time, Stratton[8] formulated a very similar model that included two additional assumptions regarding the form of the magnetic field and the size of the ion Larmor radius. These additional assumptions allowed him to make quantitative predictions for the accelerators. These two models will be described and compared with the results of the computer simulation after a discussion of the computer simulation study to be described next.

IV. COMPUTER SIMULATION OF THE DIRECT ELECTROMAGNETIC ACCELERATOR

A. Introduction

In order to obtain a solution for the motion of the plasma in the "free fall region" of the high specific impulse accelerator, a computer was used to simulate the plasma. This simulation was achieved by representing the many ions and electrons within the accelerator by several thousand particles and by following the motion of these particles in time; i.e., by the Lagrangian technique. Time was divided into small discrete increments and the following computation procedure was used to provide a time history of the fields and particle motions.

FLOW DIAGRAM
FOR
COMPUTER SIMULATION OF A PLASMA ACCELERATOR


PRESCRIBE INITIAL DISTRIBUTION OF PARTICLES
CALCULATE ELECTRIC AND MAGNETIC FIELDS DUE TO THE DISTRIBUTION OF PARTICLES
CALCULATE SURFACE CHARGES AT THE ELECTRODES THAT SATISFY CONDITIONS PRESCRIBED FOR EXTERNAL CIRCUIT
CALCULATE TOTAL ELECTRIC AND MAGNETIC FIELDS BY SUPERPOSITION OF THE FIELDS DUE TO THE DISTRIBUTION OF THE PARTICLES AND THE FIELDS DUE TO THE SURFACE CHARGES AT THE ELECTRODES
STORE ON MAGNETIC TAPE: TIME, ELECTRIC AND MAGNETIC FIELDS, AND PARTICLE POSITIONS AND VELOCITIES
CALCULATE THE MOTION OF THE PARTICLES DURING A SMALL TIME INCREMENT USING THE EQUATIONS OF MOTION WITH THE ELECTRIC AND MAGNETIC FIELDS CALCULATED ABOVE
NO
CHECK FOR PARTICLES HITTING BOUNDARIES
YES
REFLECT PARTICLES HITTING AN INSULATOR WITH A PRESCRIBED ACCOMMODATION
ABSORB ELECTRONS HITTING THE ANODE
NEUTRALIZE IONS HITTING THE CATHODE
INCREMENT TIME AND CALCULATE THE NEW DISTRIBUTION OF PARTICLES
REPEAT FOR EACH INCREMENT IN TIME

A lucid description of the Lagrangian technique was given by Buneman and Dunn in reference 9. The technique has been previously employed primarily to study electrostatic waves in a one- or two-dimensional, low density, boundless plasma or to study the ion trajectories and electron neutralization within ion guns. In both applications, the self-induced magnetic field has been neglected. A survey of the applications of this technique is given in reference 10. The computer simulation presented herein differs from the majority of previous applications in that boundaries and self-induced magnetic fields are included, the domain is an axisymmetric space rather than a one- or two-dimensional cartesian space, and the simulated plasma densities are much greater.

B. Accelerator Configurations, Boundary, and Initial Conditions

The two accelerator configurations used in the computer simulation are shown in figure 9 and were selected to simulate the accelerators being tested by Cheng[2] (shown in figure 3) and by Clark and Jahn[4] (shown in figure 4). Since detailed experimental measurements by Clark and Jahn showed that the plasma properties are axisymmetric, axisymmetry was assumed for the computer simulation.

The insulator extending downstream of the anode on the models for the computer simulation were added to simplify the mesh configurations (shown in figure 10) for the computer solution of the electromagnetic fields. The insulator could be moved out to a larger diameter with new mesh points added to fill the added space. However, the program presented uses all of the word memory of the computer which was employed; so if more mesh points are used, the number of ions and electrons that simulate the plasma would have to be reduced or a larger computer employed. Even though the added insulator causes some reduction in performance, the solutions are adequate to display the significant qualitative features of the accelerator and to check the computer simulation techniques.

It was assumed that a constant potential was applied between the electrodes and that the power supply was electrically floating. In the computer simulation these two conditions were used to calculate the surface charges on the electrodes at each time step. The condition that the power supply be electrically floating combined with the conservation of charges gives the requirement that the net transfer of electrons from the external circuit to the plasma (via the electrodes) must be the same as the net change in the total charge of the external circuit. In the computer simulation, the sum of the electrode surface charges was required to equal the net transfer of the electrons from the electrodes to the plasma. The condition that the potential difference between the electrodes be constant gives the requirement for the difference between the surface charges of the electrodes. These two requirements are sufficient to uniquely determine the electrode surface charges. Note that the floating power supply condition automatically assures that for steady state operation, the plasma leaving the accelerator will be neutral. If the electron flow rate from the circuit-accelerator combination is greater than the ion flow rate, then an excess of ions builds up in the circuit-accelerator combination increasing the electric potential until the electron and ion flow rates become the same.

No magnetic field was applied -- the only magnetic field is that induced by the particle motions. Since the model is axisymmetric with no azimuthal current, the induced magnetic field is in the azimuthal direction only.

A neutral gas was assumed to be injected through the insulator walls surrounding the cathode and the gas was arbitrarily assumed to become ionized in the regions shown in figure 9. (A simulation for Clark and Jahn's accelerator where the ionization region was assumed to be at the anode surface - considerably different than the region shown in figure 9 - illustrated that the location of the ionization region influences the efficiency of the accelerator somewhat, as described in a later section entitled "Discussion", but it did not change the qualitative features described in a later section entitled "Primary Results of the Computer Simulation"). The neutral particles were not included in the simulation -- rather electron-ion pairs were created in the assumed ionization region at a rate corresponding to the mass flow rate assuming 100% ionization of the gas. (Note that the gas can be assumed to be completely ionized even though the hydrogen ions are essentially collisionless in the low density region because the cross section for ionization of the hydrogen atom is much larger than the energy exchange cross sections for high energy ion-ion and electron-ion encounters. For example, the cross section for ionization of the hydrogen atom given by Jahn[11] is greater than 10^{-17} cm^2 whereas the ion-ion cross sections for 10^4 ev ions -- the energy corresponding to a specific impulse of 10^5 seconds -- is approximately 10^{-20} cm^2.) The simulations were found to be insensitive to the assumed thermal velocity of the newly created ions and electrons as long as the assumed thermal energy was much less than the energy of the ions at the accelerator exhaust.

The current drawn from the electrode was assumed to be proportional to the axial electric field at the cathode, and a high axial field was assumed to induce a very large current to simulate a high electrical conductivity near the cathode. The currents did not become excessive because the axial electric fields became small as described in a later section entitled "Discussion".

When the particles hit the boundaries, the encounters can be treated arbitrarily as specular, diffuse, or any combination of these. For the calculations to be presented, the particles were arbitrarily assumed to lose half of their energy with the angle of reflection equal to the angle of incidence. The exceptions are that the electrons were absorbed at the anode and the ions were considered to recombine when striking the cathode. Except for the wall encounters, the plasma was assumed to be collisionless; no hard collisions between particles were included.

The simulation was started with the accelerator void of particles and with the simultaneous initiation of mass flow and current. The computer simulation passed through a transient phase and then came to a steady state operation. The simulation was formulated to represent the steady state phase; the transient phase was not simulated accurately--rather the simulation was started with an initial condition and allowed to progress in time merely as a convenient means for obtaining the steady state operation. The modification of the program that would be required to simulate the transient phase properly is described in a later section entitled "Details of the Simulation Technique".

C. Equations of Motion

The motion of each particle is calculated using approximations to the following exact equations of motion.

$$d\vec{v} = Z\frac{e}{m}(\vec{E} + \vec{v} \times \vec{B})\, dt$$

$$d\vec{x} = \vec{v}\, dt$$

where $\vec{v}$ = velocity, m/s

Z = number of charges on the particle (= -1 for electrons)

e = charge of singly ionized ion, coulombs

m = mass of the particle, kg

$\vec{E}$ = electric field, V/m

$\vec{B}$ = magnetic field, tesla

t = time, s

$\vec{x}$ = position vector, m

The approximations employed are:

(1) The differential equations are approximated by finite difference equations.

(2) The azimuthal velocity is neglected. (The velocities induced by electromagnetic acceleration are much greater than the thermal velocities of the injected particles, and these induced velocities are in the axial and radial direction only)

With these approximations, the equations of motion as derived by Hockney[12] become:

$$v_{z_{new}} = \frac{1-\theta^2}{1+\theta^2} v_{z_{old}} + \frac{2\theta}{1+\theta^2} v_{r_{old}} + \frac{\eta}{1+\theta^2} E_z + \frac{\eta\theta}{1+\theta^2} E_r$$

$$v_{r_{new}} = \frac{-2\theta}{1+\theta^2} v_{z_{old}} + \frac{1-\theta^2}{1+\theta^2} v_{r_{old}} - \frac{\eta\theta}{1+\theta^2} E_z + \frac{\eta}{1+\theta^2} E_r$$

$$r_{new} = r_{old} + v_{r_{new}} \Delta t$$

$$z_{new} = z_{old} + v_{z_{new}} \Delta t$$

where $v_{z_{new}}$ = axial velocity at the new time, m/s

$v_{z_{old}}$ = axial velocity at the previous time, m/s

$v_{r_{new}}$ = radial velocity at the new time, m/s

$v_{r_{old}}$ = radial velocity at the previous time, m/s

r_{new} = radial position at the new time, m

r_{old} = radial position at the previous time, m

z_{new} = axial position at the new time, m

z_{old} = axial position at the previous time, m

Δt = the time increment, s

η = $\Delta t \, Z \frac{e}{m}$

θ = $\eta B/2$

These finite difference equations were formulated to yield exact solutions for two-dimensional motion when the electric and magnetic field are constant, and they are good approximations as long as the fields have relatively small variations during the time increment.

D. Field Equations

The electric and magnetic fields used in the equations of motion given in the previous section were obtained using approximations to the following exact field equations.

$$\vec{E} = -\nabla\Phi - d\vec{A}/dt$$

$$\vec{B} = \nabla \times \vec{A}$$

$$\Phi = \int_{\text{all space}} \frac{\rho}{4\pi\epsilon R} \, d\text{Volume}$$

$$\vec{A} = \int_{\text{all space}} \frac{\mu\vec{J}}{4\pi R} \, d\text{Volume}$$

where $\vec{E}$ = electric field, V/m

Φ = electric potential, V

$\vec{A}$ = magnetic potential vector, Weber/m

$\vec{B}$ = magnetic field, tesla

ρ = charge density, coulombs/m^3

$\vec{J}$ = current density, A/m^2

ϵ = electric permittivity, farad/m

μ = magnetic permeability, N/A^2

R = distance between the point of measurement of Φ or $\vec{A}$ and the location of the charge or current element

The approximations employed for the field equations are:

(1) The plasma and field properties are assumed to be axisymmetric.

(The experimental measurements by Clark and Jahn[4] showed that the properties were axisymmetric in their accelerator)

(2) The plasma and field properties are assumed to attain a steady state. (The experimental measurements by Clark and Jahn[4] showed that the properties attained a steady state)

(3) The fields are assumed to be smooth enough to be approximated by linear interpolation of field values calculated at relatively few mesh points within the accelerator.

Using the first approximation, one can obtain the magnetic field from Ampere's circuital law:

$$B_\theta = \mu I/2\pi r$$

where r = radial distance from the axis of symmetry, m

I = the current that passes through the circle formed by rotating the point of measurement about the axis of symmetry, A

From the second approximation, the electric field is given by:

$$\vec{E} = -\nabla\Phi$$

where the electric potential, using the third approximation, is given by:

$$\Phi_m = \sum_{n=1}^{N} C_{mn} \rho_n$$

where Φ_m = the electric potential at the mth mesh point

C_{mn} = the electric potential at the mth mesh point induced by a unit charge in the nth mesh cell

ρ_n = the charge density in the nth mesh cell

N = the total number of mesh cells.

The mesh configurations are shown in figure 10. Note that the mesh points are actually circles around the axis of symmetry and the mesh cells are actually toroids around the axis of symmetry. The detailed technique for calculating the electric potentials at the mesh points is described in the next section.

E. Details of the Simulation Technique

The computer simulations were obtained using a modest size computer -- an IBM 7094 with a 32K word direct access core -- with the computer program given in the appendix. The primary purpose of this work is to introduce and illustrate simulation concepts and to check the qualitative features of the simple models proposed by Stratton and the author rather than to present highly detailed solutions. Nevertheless, the computer program presented in the appendix can easily be modified to give additional detail and more accuracy with the larger computers and with more computation time.

The simulation techniques employed in the computer program that have been previously employed by others and that are explained by Buneman and Dunn[9] are as follows. The many electrons and ions of the plasma are represented by a far fewer number of representative particles -- up to 2000 electrons and 2000 ions for the programs presented in the appendix. The electric and magnetic fields were calculated from the particle distributions only at discrete points within the domain -- on the mesh points shown in figure 10 for the programs presented. The fields between these points were obtained by linear interpolation. A highly reduced ion-electron mass ratio was used -- for the calculations presented herein, usually a ratio of 10 to 1. These techniques were checked for this problem as follows. Tests wherein the number of particles used to represent the plasma was varied indicated that several thousand particles is an adequate statistical sample to use for this two-dimensional problem. Tests with the mesh size reduced to half the size shown in figure 10 indicated that the course mesh was adequate because the fields were smoothly varying for this problem. Tests with mass ratios varying from 2 to 32 indicated that the steady state solutions were only slightly sensitive to a variation of the mass ratio from 2 to 8 and insensitive to the variation of mass ratio from 8 to 32.

The reduced mass ratio is used in the computer simulation to reduce the computation time.' The time step used in the simulation must be smaller than the inverse of the plasma frequency in order that the numerical solution be stable.

$$\Delta t < \frac{1}{\omega_p}$$

where Δt = the time step used in the simulation, s

ω_p = the plasma frequency = $\left(\frac{n_e e^2}{\epsilon_o m_e}\right)^{1/2}$, radians/s

n_e = electron number density, electrons/m^3

e = charge of a singly ionized ion, coulombs

ϵ_o = electric permittivity of free space, farads/m

m_e = electron mass, kg

(Note that the plasma frequency is a natural oscillation frequency of the electrons, analogous to the natural oscillation frequency of a spring-mass system; hence, one can see the physical reason why the computations become unstable if the time step exceeds the inverse of the plasma frequency). By increasing the electron mass to one-tenth the ion mass, the time step can be increased and the computation time reduced more than one order of magnitude. (Note that the increased electron mass changes the plasma frequency; so if one wanted to simulate the electron fluctuations in an unsteady plasma, one would have to use the proper electron mass.)

A new technique for simulating the steady flow of high density plasmas was examined and employed for the simulations presented herein. Since the minimum time step is inversely proportional to the square root of the electron number density, as shown above, the computation time is proportional to the square root of the electron number density; therefore, the simulation of dense plasmas using the Lagrangian technique described above had not been attempted previously. However, the simulations presented herein for the steady dense plasma flows within the accelerators were obtained with reasonable computation times by artificially increasing the electric permittivity, which reduces the plasma frequency and reduces the computation time. (Note that this technique is similar to artificially increasing the electron mass in that both artificially reduce the plasma frequency. Neither technique can be used for simulating plasmas where the electron fluctuations must be considered).

Some understanding of the effect of artificially increasing the electric permittivity can be obtained by considering the flow of an inviscid plasma through an insulated channel. When an electric field is applied transverse to the flow, the electrons and ions are deflected in opposite directions toward the channel walls until plasma sheaths are formed which counterbalance the applied electric field. The particle trajectories are not altered greatly by the applied electric field if the relative shift of electrons and ions to form the plasma sheaths is small. The distance the electrons and ions shift to counterbalance the applied field is

$$\delta \approx \left(\frac{\epsilon_o e\Phi}{n_e e^2}\right)^{1/2}$$

where δ = the distance the electrons and ions shift toward the walls to form the plasma sheath, m

ϵ_o = electric permittivity of free space, farads/m

e = charge of a singly ionized ion, coulombs

Φ = the electric potential applied across the channel, V

n_e = electron number density, electrons/m^3

The deflection of the individual particles is very small compared to the channel width, W, when

$$\left(\frac{\epsilon_o \Phi}{en_e W^2}\right)^{1/2} << 1$$

Note that this condition is the same as the condition that the applied potential be much smaller than the potential that would be produced by removing all of the ions from the channel, or, equivalently, that the relative charge density, $\hat{\rho} \equiv \frac{|\rho_i - \rho_e|}{\rho_e}$, be much less than unity. Artificially increasing ϵ_o permits the electrons and ions to shift more, creating an artificially large relative charge density. However, little error in the particle trajectories is produced by increasing ϵ_o as long as the above condition is met; i.e., as long as the relative charge densities are small compared to unity.

The ion and electron number densities within the accelerators simulated are of the order of 10^{15} particles/cm^3. However, the net charge density, the difference between the ion and electron number densities, is of the order of 10^9 charges/cm^3. Therefore, the relative charge density, $\hat{\rho}$, is of the order of 10^{-6}. Although a frequently used technique for treating plasmas with small relative charge densities is to artificially set $\hat{\rho}$ identically equal to zero, assuming perfect neutrality, the technique employed herein was to artificially increase the electric permittivity, ϵ_o, permitting $\hat{\rho}$ to increase from the order of 10^{-6} to the order of 10^{-1}. Tests that were performed varying the electric permittivity illustrated that the ion and electron distributions adjusted slightly to give very nearly the same electric potential despite the variation in electric permittivity. (Only slight adjustments of the particle trajectories were required to give these adjustments in electron and ion distributions.) In fact, the qualitative features of the simulation were satisfactorily given even with the relative charge densities as high as 1/3.

Butler, Cook, and Morse[13] have assumed complete neutrality, $\hat{\rho} = 0$, in order to obtain a computer solution of a coaxial accelerator such as Cheng's. Whenever complete neutrality is assumed, the electric field cannot be calculated from the charge distributions -- rather, it must be calculated from the motion of the plasma. Butler, Cook, and Morse calculated the particle motion by assuming the plasma was frozen to the magnetic field lines and by treating the magnetic pressure as the dominant accelerating force. (The electric fields are not even used in this type of calculation even though in actuality the ions receive all of their energy from the electric fields.) Their technique should take less computing time for simulating plasmas that are frozen to field lines. However, it is shown in a later section entitled "Primary Results of the Computer Simulation" that the plasma is not frozen to the magnetic field lines within Cheng's accelerator.

The potential field at each mesh point was calculated by superimposing the potentials induced at the mesh point by the surface charges and by the charges within all of the mesh cells rather than calculated from the technique developed by Hockney[12,14]. (The charges were assigned to the mesh cells using the method described by Birdsall and Fuss[15].) The potential induced on all mesh points by a unit charge in one mesh cell or by a unit surface charge at an electrode was calculated once at the start of the program, and these inverse capacitances were merely multiplied by the actual charge in each mesh cell or the actual surface charge to calculate the electric potentials at each time step. This technique is particularly suitable for problems with mixed boundary conditions, such as for intermittent conductors and insulators, and for problems with less than 1000 mesh points. For problems with more mesh points and with the special boundary conditions cited by Hockney, the technique he presented would take less computer time. The solution for the potential fields using the technique just described for the mesh configuration shown in figure 10 required 5 milliseconds on an IBM 7094.

The steady magnetic field was easily determined for this axisymmetric configuration using Amperes Circuital Law.

$$B_\theta(r,z) = \frac{\int_0^r J_z(\xi,z)2\pi\xi d\xi}{2\pi r}$$

The current density was calculated at each mesh point from the particle velocities for the particles inside the mesh cell.

To calculate unsteady fields, one could use the magnetic potential vector with the equation

$$\nabla^2\vec{A} = \mu\vec{J}$$

and the same program for the solution of Poisson's Equation that is employed to solve for the electric potentials could be used for the solution of the magnetic potentials. Such calculations as these would be necessary in order to properly simulate the transient phase of the accelerator operation.

The computation times required to achieve steady state solutions for the plasma accelerators were approximately 60 minutes on the IBM 7094 computer using several thousand representative particles and using the mesh configuration shown in figure 10.

F. Display Techniques

A computer program was developed to display the calculated particle motions, the electric pc.ential, and the magnetic field simultaneously on a cathode ray tube. The cathode ray tube display was recorded on motion pictures; a typical frame of the motion pictures is shown in figure 11. Since the model is assumed axisymmetric, the properties on any plane through the axis are the same. The motion of the particles is shown on one such plane in the center figure, which is schematically shown as a cross section of the accelerator. The same plane has been translated to the upper left to show the electric potential and to the lower right to show the magnetic field. Since no magnetic field is applied, the model is axisymmetric, and the azimuthal velocities are assumed to be zero, the only component of the magnetic field is azimuthal; the absolute magnitude of this azimuthal field is the value plotted. The motion pictures permit one to observe the interaction between the particles and the fields. One can see how the fields accelerate the particles, how the particle motion changes the charge density and current distributions, and how the changing distributions modify the fields that accelerate the particles.

The calculations for the simulation and the display were not done simultaneously. The particle positions and the electromagnetic fields were calculated on the IBM 7094 and these quantities were stored on magnetic tape. The information on the magnetic tape was then processed and plotted with a CDC 3200 computer-CDC 280 cathode ray tube combination. This computer-cathode ray tube combination produced the frames illustrated on figure 11 at approximately 2 frames/sec. The computation time required to plot the simulation was approximately 1/10 the time to generate the information with the IBM 7094.

Because of the large volume of information processed, special care was taken to minimize tape read and write times, and parallel operations were performed where possible. For example, the floating numbers generated for the particle positions and for the electromagnetic fields were converted to 12 bit integers, and three such integers were packed into each 36 bit IBM 7094 word prior to recording the information onto magnetic tape. The CDC 3200 was programmed so that it plotted the information for one frame, processed the information of a second frame, and read the information for a third frame simultaneously.

G. Primary Results of the Computer Simulation

The more important qualitative features observed in the computer simulation movies are as follows. Despite the difference in shape between Cheng's accelerator and Clark and Jahn's accelerator, the same features are noted for both. The self-induced magnetic field causes a large fraction of the electrons emitted from the cathode to be deflected from a straight path to the anode toward a path parallel to the axis of the accelerator. The increase in electron density near the axis in turn lowers the electric potential along the axis such that the electric field in the region of ionization becomes mainly radial as shown in figure 12. A large fraction of the ions accelerated toward the axis by the radial field are deflected by the induced magnetic field toward the accelerator exhaust along a path similar to the one shown in figure 13.

Most of the electrons that reach the anode are the electrons released by the ionization of the gas, rather than electrons that originate at the cathode. The majority of the electrons that originate at the cathode leave the accelerator with the ions that are deflected to the accelerator exhaust forming the neutral plasma stream that leaves the accelerator. The radial current is not everywhere mainly electron current as in the MPD arjet -- rather an appreciable part of the radial current is carried by ions. With the ionization regions as shown in figure 9, approximately half of the total radial current is ion current.

The simulation shows that, for this problem, two frequently used concepts must be modified. The concepts are (1) that the $\rho_i \vec{E}$ force cancels the $\rho_e \vec{E}$ force for a neutral plasma, and (2) that plasmas with infinite conductivity are frozen to magnetic field lines. Although the plasma is nearly neutral, the $\rho_i \vec{E}$ force is not strictly cancelled by the $\rho_e \vec{E}$ force as in a highly collisional plasma because in the direct electromagnetic accelerator the electron and ion gases can easily slip through one another. Rather, the $\rho_i \vec{E}$ force is balanced partially by inertia forces and partially by the $\vec{v}_i \times \vec{B}$ force. Figure 14 shows an ion trajectory for an ion starting from rest in a uniform electric field and a decaying magnetic field -- this trajectory is similar to the ion trajectories observed in the numerical simulation. As the ion begins its motion, the $\rho_i \vec{E}$ force is balanced almost entirely by the inertia force whereas during the last part of the trajectory shown in figure 14, the $\rho_i \vec{E}$ force is balanced by the $\vec{v}_i \times \vec{B}$ force. The $\rho_e \vec{E}$ force on the electrons is balanced almost entirely by the $\vec{v}_e \times \vec{B}$ force since the inertia of an electron is so much smaller than the inertia of an ion. The typical electron trajectory for the same fields as for the ion is also shown on figure 14.

Although the electrical conductivity in the simulation is effectively infinite because there are no particle-particle collisions included in the simulation, the fluid is not frozen to the magnetic field.

With the reference frame fixed in the accelerator, the magnetic field lines are stationary and the particles are accelerated through them on trajectories such as shown in figure 13. Furthermore, since the velocity of the ions and electrons vary in direction and magnitude throughout the accelerator, there is no transformation of reference frames possible where the electric field could be transformed to a moving magnetic field in which the fluid would be frozen to the magnetic field lines. One can deduce from the simulation that it is the inertia terms that cause the slippage of plasma through the field lines. Particles with finite inertia may rotate around the field lines with a finite gyro radius. In the computer simulation, the gyro radius for the ions is approximately equal to the size of the accelerator and to the size of the volume that contains a large magnetic field.

H. Comparison of the Computer Simulation with the Simple Models

The essential features that were incorporated as a priori assumptions in the simple models proposed by Stratton[8] and the author were corroborated by the computer simulation. These assumed features are: (1) the self-induced magnetic field channels the electrons from the cathode along the axis, causing a dip in the potential field along the axis -- the ions then receive their energy from the resulting predominately radial field; and (2) the self-induced magnetic field converts the radial velocity obtained by the ions from the electric field to an axial velocity as shown in figure 13. These features are not assumed for the computer simulation -- rather, the major assumptions for the computer simulation are that the dominant acceleration mechanism can be simulated with a collisionless model and that sufficient particles are treated in the simulation to comprise a reasonable statistical sample. Because the two features are found as a result of calculating the self-consistent fields and particle motions in the computer simulation, the simulation corroborates the a priori assumptions of the simple models.

These features, when stated in equation form are:

$$\frac{m_i v_i^2}{2} = C_1 Ze\Phi \qquad (1)$$

the ions can receive energy only from the electric field where

m_i = mass of the ion , kg

v_i = the exit velocity of the ion, m/s

C_1 = a proportionality constant less than 1

Φ = the potential applied to the accelerator, V

Z = number of charges on each ion

e = charge of a singly ionized ion, coulombs

$$v_{i_{axial}} = \int_{\text{ion path}} \frac{Ze\, v_{i_{radial}} B_\theta}{m_i} dt = \int_{\text{ion path}} \frac{ZeB_\theta}{m_i} dr \qquad (2)$$

the radial velocity received from the electric field is converted to axial velocity by the magnetic field where

B_θ = aximuthal magnetic field, tesla

t = time ,s

r = radial position within the accelerator ,m

Stratton[8] and the author further proposed that, for efficient performance, the number of neutral particles injected into the accelerator times the average number of charges per ion should be approximately the same as the number of electrons emitted at the cathode. This hypothesis was not studied with the computer simulation; however the following argument is presented to support its validity: For steady operation the number of electrons leaving the accelerator must equal the number of positive charges (number of ions times the average number of charges per ion) leaving the accelerator. Forcing the number of electrons emitted at tne cathode to be much larger would merely force many of the electrons across the potential to the anode, and since the electrons cannot effectively transfer their energy to the ions, this would merely result in additional heating to the anode.

Stated in equation form this is written

$$\dot{m} = \frac{C_2 m_i I}{Ze} \qquad (3)$$

where

$\dot{m}$ = mass flow rate of injected gas ,kg/s

I = current ,A

C_2 = proportionality constant (approximately equal to unity)

If the relationship between the current and the magnetic field is known or assumed, then equation (2) becomes

$$v_i = C_3 I \qquad (4)$$

where C_3 = proportionality constant.

In order to predict quantitative characteristics for direct electromagnetic accelerators, the proportionality constants C_1, C_2, and C_3 must be known or assumed. Since the potential the ions fall through is of the same order as the applied potential, C_1 must be of the order of unity. C_2 must be of the order of unity for efficient operation, and was set equal to one for the computer simulation by setting the rate of ionization equal to the rate of electron emission from the cathode. In order to obtain C_3, Stratton assumed the magnetic field to be

$$B_\theta = \frac{\mu I}{2\pi r} \qquad (5)$$

where μ = magnetic permeability, and assumed the ion gyro radius is everywhere equal to the geometrical radius from the axis of the accelerator. With these assumptions, Stratton obtained a value of 20 for C_3. On the other hand, the computer simulation yields a value of C_3 of approximately 15 for the model of Clark and Jahn's accelerator and approximately 6 for the model of Cheng's acceleratoı. The magnetic field obtained from the computer simulation, shown in figure 11, does not compare well with Stratton's assumption given by Equation 5, and the typical ion path shown in figure 13 does not agree with Stratton's assumption for the ion gyro radius. Nevertheless, the value of C_3 that Stratton obtained from these assumptions are within the same order of magnitude as the values obtained from the computer simulation.

I. Discussion

From the collisionless model of the direct electromagnetic accelerator, one can deduce (1) that the energy of the accelerated ions, in electron volts, cannot exceed the voltage applied to the accelerator, and (2) that the optimum performance would occur with no applied magnetic field. These are in contrast to the electric accelerators operating in the MPD regime where experimental measurements have shown that the accelerated ions can receive an energy in electron volts greater than the voltage applied to the accelerator[16] and wherein an applied magnetic field can improve the performance[17]. When charged particles receive energy only from steady fields, they cannot obtain a higher energy than that of falling through the maximum difference in potential. For the collisionless model, the energy of the ions leaving the accelerator will equal in electron volts the difference in potential between the point of ionization and the potential outside of the accelerator exhaust. The maximum thrust will occur when all of the energy the ion receives falling through the potential is converted to axial velocity. The production of an azimuthal velocity would merely shift part of the axially directed energy to a rotational energy. Since a steady axial or radial applied magnetic field produces rotation about the axis but cannot add energy to the ions, the rotational energy would be subtracted from the axially directed energy resulting in a reduced thrust.

The coulomb cross section for high energy collisions of charged particles with ions becomes less than the hard sphere cross section of the ions if the ions have electron shells. For example, the ion-ion cross section for hydrogen ions with 10^4 ev energy is of the order of $10^{-20}cm^2$ whereas the hard sphere cross section for ions with electron shells is of the order of 10^{-15} cm^2. Therefore, although the hydrogen ion mean free paths may be large compared to the size of the accelerator for the high specific impulse accelerators, the mean free path of other ions would be of the order of the size of the accelerators at ion number densities of $10^{15}cm^3$. Therefore, the collisionless model would not apply rigorously for gases other than hydrogen. Nevertheless, the ions would still receive most of their energy directly from the electric field, and the collisionless model may still predict the accelerator characteristics satisfactorily. (The influence of the atomic weight and ion charge on the characteristics in the collisionless model will be discussed in the next chapter entitled "Performance Predictions for the Direct Electromagnetic Accelerator".)

The voltage-current characteristics given by the computer simulations are shown in figure 15. It was assumed that large axial electric fields at the cathode would induce very large currents from the cathode, simulating high electrical conductivity in and near the cathode. The simulation was started with an arbitrarily low current and, as the simulation progressed in time, the current increased to an asymptotic value. As the current increased, the self-pinch caused the axial electric field near the cathode to decrease to a small value as shown in figure 12 so that the asymptotic currents do not become excessively large. Since there are no particle-particle collisions in the simulation, the electrical conductivity is essentially infinite. The radial electron current throughout the accelerator, however, is limited by the fact that the electric fields are balanced by the $\vec{v}_e \times \vec{B}$ force on the electrons. The ion currents are limited by inertia forces in addition to the partial balancing of the electric field with the $\vec{v}_i \times \vec{B}$ force.

Although the computer simulation was developed primarily for comparison with the qualitative assumptions made for the simplified models, it may be used to simulate the plasma flows in accelerators of various types to determine which configurations will be the most efficient. For example, introducing the gas at the anode and forcing it to be ionized in the anode sheath would force the ions to fall through a greater potential than if the gas were ionized nearer to the cathode. Therefore, an accelerator with the gas introduced at the anode should produce higher thrust than one with the gas introduced near the cathode. A simulation was performed with the ionization region assumed to be near the anode rather than distributed according to figure 9, and an increase in thrust was noted.

The computer simulation could be extended to include cases with applied magnetic fields. This extension would require additional computer storage space in order to store the values of the radial and axial components of the magnetic field and to store the azimuthal velocities of the particles. The azimuthal force equation would have to be added in order to calculate these azimuthal velocities, and the azimuthal currents would have to be computed from these velocities in order to calculate the induced axial and radial magnetic fields. These additional calculations would increase the computation time by an estimated 20%.

The values of velocity, thrust, and voltage obtained with the computer simulation of Cheng's accelerator agree with the experimental measurements of Cheng to within the accuracy of the experimental measurements.

The current distribution within the simulation is similar to the current distributions measured by Clark and Jahn[4] in that current locps protrude downstream of the anode. However, further comparison is not yet possible because Clark and Jahn have not yet reported values of velocity, thrust, or voltage within their high current accelerator

V. PERFORMANCE PREDICTIONS OF THE DIRECT ELECTROMAGNETIC ACCELERATOR

A. Dimensionless Parameters

When the equations used in the computer simulation program (given in the appendix) are made dimensionless, the significant parameters that emerge are

$$\left(\frac{Ze}{m_i}\right)\Phi\left(\frac{\Delta t}{L}\right)^2$$

$$\mu\left(\frac{Ze}{m_i}\right) I \left(\frac{\Delta t}{L}\right)$$

$$\mu\left(\frac{Ze}{m_i}\right)^2 \dot{m} \left(\frac{\Delta t}{L}\right)$$

where

Z = the number of charges on each ion

e = the charge of a singly ionized ion, coulombs

m_i = mass of the ion, kg

Φ = potential applied across the electrodes, V

Δt = incremental time step between calculations, s

L = characteristic dimension of the accelerator, m

μ = magnetic permeability, N/A^2

I = current, A

$\dot{m}$ = mass flow rate, kg/s

The parameters above are fixed for each numerical simulation, but the simulation applies to any combination of applied voltages, currents, mass flow rates, accelerator sizes, and time scales that give the same dimensionless parameters. For example, one simulation that represents an applied potential of 1000 volts, a current of 20,000 amperes, and a hydrogen flow rate of .2 g/s also simulates the same accelerator with an applied potential of 9000 volts, a current of 60,000 amperes, and a hydrogen flow rate of .6 g/s. The movie of the simulation also represents either case, but the incremental time step for the second case is 1/3 the incremental time step for the first case so the time dilitation of the movie is different for the two cases.

From these dimensionless parameters, the variation of current, mass flow rate, specific impulse, and thrust as a function of the applied voltage, the atomic weight, and the number of charges of each ion for a given accelerator are as follows:

I	$\propto (\Phi A/Z)^{1/2}$
$\dot{m}$	$\propto \Phi^{1/2}(A/Z)^{3/2}$
specific impulse	$\propto (\Phi Z/A)^{1/2}$
thrust	$\propto \Phi A/Z$
	where A = atomic weight

B. Predicted Performance Charts

The performance obtained from the computer simulation for Clark and Jahn's accelerator and for Cheng's accelerator are shown in figures 16 and 17. For voltages below the limit marked "collisional plasma" on figure 16, the collisionless approximation is not expected to be realistic, and for voltages above the limit marked "relativistic electrons", one would have to include the relativistic effects in the computer simulation.

The values given on figure 16 are independent of the size, but the thrust density shown on figure 17 depends on the cross sectional area of the accelerator. The cross sectional area of Cheng's accelerator is approximately 10 cm^2 whereas Clark and Jahn's is approximately 100 cm^2. This difference in cross sectional area is the primary reason the curve for Clark and Jahn's accelerator is so far below the curve for Cheng's accelerator.

When the assumed location for the region of ionization was varied with the applied voltage held constant, the current, mass flow rate, velocity, thrust, and specific impulse varied less than a factor of two. Therefore, these predictions are considered to be good enough for order of magnitude estimates despite the fact that the location for the region of ionization is unknown and had to be assumed.

VI. CONCLUSIONS

The ions within the very high specific impulse accelerators (such as Cheng's coaxial accelerator) are accelerated directly by the electromagnetic fields rather than by collisions with the electrons. For this regime of direct electromagnetic acceleration, the present MPD arcjet theories are inapplicable.

A collisionless plasma model is recommended to represent the region where the ions are receiving their energy directly from the electromagnetic fields. The Lagrangian computer simulation technique appears to simulate satisfactorily the plasma within the collisionless region of these direct electromagnetic accelerators. The values of velocity, thrust, and voltage obtained with the computer simulation of Cheng's accelerator agree with the experimental measurements of Cheng to within the accuracy of the experimental measurements.

Results from the computer simulation show that the simple particle models proposed by Stratton and the author contain the essential features for describing the acceleration mechanism within direct electromagnetic accelerators. Order of magnitude estimates of the performance characteristics of the direct electromagnetic accelerator obtained from the computer simulation are in agreement with the predictions of Stratton's collisionless particle model despite the fact that some of Stratton's assumptions are not corroborated by the computer simulation.

Results of the computer simulation also illustrate that the $\rho_e\vec{E}$ force term does not strictly counteract the $\rho_i\vec{E}$ force term even though the plasma is nearly neutral. Rather, the $\rho_e\vec{E}$ force is balanced by the $\vec{v}_e \times \vec{B}$ force, and the $\rho_i\vec{E}$ force is the dominant force for adding energy to the ions. In addition, even though the electrical conductivity is essentially infinite because no particle-particle encounters are considered, the plasma is not frozen to the magnetic field lines because of the large ion inertia forces.

The direct electromagnetic accelerator should operate most efficiently with no applied magnetic field and with a mass flow proportional to the current. The specific impulse should increase in proportion to the current, and the voltage and thrust should increase with the square of the current.

REFERENCES

1. Clark, Kenn E. and Jahn, Robert G., "The Magnetoplasmadynamic Arcjet," Astronautica Acta, Vol. 13, pp 315-325, Pergamon Press Ltd., March 1967.

2. Cheng, Dah Yu, "Focussed Discharge From Pulsed Plasma Guns," Paper presented at the 1967 Annual Meeting of the Division of Plasma Physics, Austin, Texas, November 8-11, 1967.

3. Ducati, Adriano C., Muehlberger, Erich, and Giannini, G., "High Specific Impulse Thermo-Ionic Acceleration," AIAA Paper No. 64-668, AIAA Fourth Electric Propulsion Conference, August 31 - September 2, 1964, Philadelphia, Pennsylvania.

4. Clark, Kenn E. and Jahn, Robert G., "Quasi-Steady Plasma Acceleration," AIAA paper No. 69-267 AIAA 7th Electric Propulsion Conference, March 3-5, 1969, Williamsburg, Virginia.

5. Stine, Howard A. and Watson, Velvin R., "The Theoretical Enthalpy Distribution of Air in Steady Flow Along the Axis of a Direct-Current Electric Arc," NASA TN D-1331, August 1962.

6. Watson, Velvin R. and Pegot, Eva B., "Numerical Calculations for the Characteristics of a Gas Flowing Axially Through a Constricted Arc," NASA TN D-4042, June 1967.

7. Nerheim, N.M. and Kelly, A.J., "A Critical Review of the State-of-the-Art of the MPD Thrustor," AIAA Paper No. 67-688, AIAA Electric Propulsion and Plasmadynamics Conference, Colorado Springs, Colorado, September 11-13, 1967.

8. Stratton, T.F., "High Current Steady State Coaxial Plasma Accelerators," AIAA Journal, Vol. 3, Number 10, October 1965, pp 1961-1963.

9. Buneman, O. and Dunn, D.A., "Computer Experiments in Plasma Physics," Science Journal, July 1966, pp 1-8.

10. Symposium on Computer Simulation of Plasma and Many-Body Problems, NASA SP-153, April 19-21, 1967, Williamsburgh, Virginia.

11. Jahn, Robert G., "Physics of Electric Propulsion," McGraw-Hill Series in Missile and Space Technology, McGraw-Hill, 1968.

12. Hockney, R. W., "Computer Simulation of Anomalous Plasma Diffusion and Numerical Solution of Poisson's Equation," SUIPR Report No 53, Stanford University, May, 1966.

13. Butler, T. D., Cook, J. L., and Morse, R. L., "MHD Simulation of Co-Axial Plasma Flow," Proceedings of the APS Topical Conference on Numerical Simulation of Plasma, LA-3990, September 18-20, 1968.

14. Hockney, R. W., "The Potential Calculation," Proceedings of the APS Topical Conference on Numerical Simulation of Plasma, LA-3990, September 18-20, 1968.

15. Birdsall, C. K. and Fuss, D., "Cloud-In-Cell Computer Experiments in Two and Three Dimensions," Proceedings of the APS Topical Conference on Numerical Simulation of Plasma, LA-3990, September 18-20, 1968.

16. Grossman, William, Jr. and Hess, Robert V., "Experiments with a Coaxial Hall Current Plasma Accelerator," AIAA Journal, Vol. 3, No. 6, June 1965.

17. John, Richard R., Bennett, Stewart, and Jahn, Robert, "Current Status of Plasma Propulsion," AIAA Paper No. 66-565, AIAA Second Propulsion Joint Specialist Conference, Colorado Springs, Colorado, June 13-17, 1966.

APPENDIX. COMPUTER PROGRAMS FOR SIMULATING THE DIRECT ELECTROMAGNETIC ACCELERATORS

These computer programs were written in FORTRAN IV and were run on the IBM 7094 DCS system at Ames Research Center, NASA. The programs are located as follows:

PROGRAM PAGE

```
CMP04T1 *** DIRECT ELECTROMAGNETIC ACCELERATOR *** VAL WATSON                  2696
C
C
C
C*********************************************************************
C
C        MAIN PROGRAM FOR THE COMPUTER SIMULATION OF CLARK AND JAHNS
C                       AXISYMMETRIC PLASMA ACCELERATOR
C
C*********************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(2000), RE(2000), ZI(2000), ZE(2000), VRI(2000), VRE(2000),
     1     VZI(2000), VZE(2000)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(5,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)
      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1     NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST
C ************************************************************
C
C
      FIRST = 1
C
C
C *** INITIAL VALUES ***
C
C
      CALL INITL
C        SUBROUTINE INITL INITIALIZES ALL OF THE COMMON VARIABLES
C
      CLOCK4 = 0.0
      CURRNT = AMPSE
C
C
C *** TIME LOOP ***
C
C
      DO 200 TSTEP=1,NTSTEP
      CALL FIELDS
C        SUBROUTINE FIELDS EVALUATES THE ELECTRIC AND MAGNETIC FIELDS
C        FROM THE PARTICLE DISTRIBUTIONS
C
C
C *** OUTPUT ON TAPE FOR FUTURE PLOTTING ***
C
```

```
      IF(OTTINC.EQ.0) GO TO 80
      IF(TSTEP.EQ.NTSTEP) GO TO 40
      IF(OTTIND.GT.0) GO TO 80
      OTTIND = OTTINC
   40 CALL OTTPT
C         SUBROUTINE OTTPT STORES THE PARTICLE POSITIONS AND THE FIELDS ON
C         MAGNETIC TAPE
C
   80 OTTIND = OTTIND - 1
      IF(OUTIND.GT.0) GO TO 100
      OUTIND = OUTINC
C
C *** WRITTEN OUTPUT ***
C
      CALL OUTPT
C         SUBROUTINE OUTPUT PRINTS THE VALUES FOR THE LOCAL FIELDS,
C         DENSITIES, AND CURRENTS
C
      CLOCKP = CLOCK4
      CALL CLOCK(CLOCK4)
      CLOCKS=CLOCK4-CLOCKP
      WRITE(6,700) CLOCKS
  700 FORMAT(1H0,20H TIME FOR OUTPT IS   ,1PE10.3)
  100 OUTIND = OUTIND - 1
C
C
      T = T + DT
      DT = DT*EX
      IF(TSTEP.GT.1) CALL TRAJ
C         SUBROUTINE TRAJ CALCULATES THE PARTICLE VELOCITIES AND POSTIONS
C         AT EACH TIME STEP
C
C
C *** MAKING THE CURRENT PROPORTIONAL TO THE AXIAL ELECTRIC FIELD AT THE
C         CATHODE
C
```

```
      AMPS =    -CURRNT * EZ(1,1) *100.0 * DZ/VOLTS
         IF(AMPS.LT.0.0) AMPS = 0.0
         AMPM = 1.5*CURRNT
         IF(AMPS.GT.AMPM) AMPS = AMPM
      DATA TAVN/40.0/
      AMPSE = (AMPS + (TAVN-1.0)*AMPSE)/TAVN
      TAVN = TAVN + 1.0
         AMPSI = AMPSE
         WRITE(6,800) AMPSE
  800    FORMAT(1H0,15H CURRENT =          ,F15.3)
C
      CALL SOURCE
C        SUBROUTINE SOURCE PROVIDES FOR INJECTION OF PARTICLES INTO THE
C        ACCELERATOR EITHER BY VOLUME IONIZATION OR FROM THE BOUNDARIES
C
      IF(NEXCD.GT.0) GO TO 400
  200 CONTINUE
      GO TO 500
  400 WRITE(6,600)
  500 CONTINUE
  600 FORMAT(10X,30H EXCEEDED 2000 PARTICLES       )
      LAST = 1
      RETURN
      END
```

```
CMP04T2 SUBROUTINE INITL               VAL WATSON                            2696
      SUBROUTINE INITL
C
C
C
C***********************************************************************
C
C        THIS SUBROUTINE INITIALIZES THE COMMON VARIABLES
C
C***********************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(2000), RE(2000), ZI(2000), ZE(2000), VRI(2000), VRE(2000),
     1     VZI(2000), VZE(2000)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(5,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)
      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1     NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NExCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST
C ************************************************************
      CALL LOCATE(1,8)
      CALL LOCATE(1,10)
C
C
C *** INPUT DATA ***
C
C
      READ(5,300) DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,AMPSE,AMPSI
      READ(5,301) NTSTEP,OUTINC,OTTINC
      WRITE(6,400) DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,AMPSE,AMPSI
     1    ,NTSTEP,OUTINC,OTTINC
C
C
C *** SET FIXED CONSTANTS ***
C
C
      EX = 1.0
      DR = 0.01
      DZ = 0.01
      PI = 3.14159
      Q = 1.6E-19
      QNET = 2.0E-9
      MUZERO = 4.0*PI*1.0E-7
      EPSO = 8.85E-12
      MASSI = 1.67E-27
      MASSE = MASSI/MASSR
C
C
C ***  SET INTERNAL VARIABLES ***
C
C
      T = 0.0
      RA=5
      ZA=5
      RC = 1
      ZC = 1
      OUTIND = 0
      OTTIND = 0
      NFILE = 0
      NFRAME = 0
      NIT = 0
      NLI = 0
      N = 0
      NET = 0
      NLE = 0
      NEXCD = 0
      NRECOM = 0
      RE(1) = 0.0
      RI(1) = 0.0
      ZE(1) = 0.0
      ZI(1) = 0.0
      VRE(1) = 0.0
      VRI(1) = 0.0
      VZE(1) = 0.0
      VZI(1) = 0.0
      DO 2 M=1,400
      NDELI(M) = 0
      NDELE(M) = 0
    2 CONTINUE
      EXTRA1 = 0.0
      EXTRA2 = 0.0
      EXTRA3 = 0.0
      NXTRA1 = 0
      NXTRA2 = 0
      NXTRA3 = 0
C
C
C *** GENERATE RADII AND VOLUMES ***
C
C
      RAD(1) = DR/4.0
      VOL(1) = DZ*PI*DR*DR/4.0
      DO 4 I=2,5
      FI = I
      RAD(I) = (FI-1.0)*DR
      VOL(I) = 2.0*PI*RAD(I)* DR * DZ
    4 CONTINUE
      RDS = RAD(5)
      LENG = 14.0 * DZ
      VOL(5) = VOL(5)/2.0
C
C
C *** FORMATS ***
C
C
  300 FORMAT(E10.3)
  301 FORMAT(I4)
  400 FORMAT(1H1, 50H +++++++ MPD ARC JET - LAGRANGE SOLUTION +++++++  /
     1     10X, 10H**INPUT**      /
     2     20X, 10H DT    =   , 1PE15.5/ 20X 10H MASSR =   , 1PE15.5/
     2     20X, 10H NAMPL =   , 1PE15.5/ 20X 10H EPSAMP=   , 1PE15.5/
     2     20X, 10H REFLC =   , 1PE15.5/ 20X 10H VOLTS =   , 1PE15.5/
     2     20X, 10H AMPSE =   , 1PE15.5/ 20X 10H AMPSI =   , 1PE15.5/
     2     20X, 10H NTSTEP=   , I4     / 20X, 10H OUTINC=  , I4   /
     2     20X, 10H OTTINC=   , I4     )
C
```

```
C
      LAST = 1
      RETURN
      END                                                                 146
```

```
CMP04TB SUBROUTINE INITL                VAL WATSON                           2696
      SUBROUTINE INITL
C
C
C
C***********************************************************************
C
C        THIS SUBROUTINE INITIALIZES THE COMMON VARIABLES FOR A
C                  CONTINUATION RUN
C
C***********************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(2000), RE(2000), ZI(2000), ZE(2000), VRI(2000), VRE(2000),
     1     VZI(2000), VZE(2000)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(5,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)
      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1     NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST
C *********************************************************
C
C
C *** INPUT DATA ***
C
C
      READ(5,301) NFILE
      CALL LOCATE(NFILE,8)
      READ(5,300) DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,AMPSE,AMPSI
      READ(5,301) NTSTEP,OUTINC,OTTINC
      WRITE(6,400) DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,AMPSE,AMPSI
     1    ,NTSTEP,OUTINC,OTTINC
C
C
C *** SET FIXED CONSTANTS ***
C
C
      EX = 1.0
      DR = 0.01
      DZ = 0.01
      PI = 3.14159
      Q = 1.6E-19
      MUZERO = 4.0*PI*1.0E-7
      EPSO = 8.85E-12
      MASSI = 1.67E-27
      MASSE = MASSI/MASSR
```

```
C

C
C ***  SET INTERNAL VARIABLES ***
C
C
      READ(8)   T,PHI,B,NUMI,NUME,NUMVI,NUMVE,NIT,(RI(I),ZI(I),VRI(I),VZ
     1I(I),I=1,NIT),NET,(RE(I),ZE(I),VRE(I),VZE(I),I=1,NET),NFRAME,QNET
      CALL SKIP(1,8)
      RA=5
      ZA=5
      RC = 1
      ZC = 1
      OUTIND = 0
      OTTIND = 0
      NLI = 0
      N = 0
      NLE = 0
      NEXCD = 0
      NRECOM = 0
      RE(1) = 0.0
      RI(1) = 0.0
      ZE(1) = 0.0
      ZI(1) = 0.0
      VRE(1) = 0.0
      VRI(1) = 0.0
      VZE(1) = 0.0
      VZI(1) = 0.0
      DO 2 M = 1,400
      NDELI(M) = 0
      NDELE(M) = 0
    2 CONTINUE
      EXTRA1 = 0.0
      EXTRA2 = 0.0
      EXTRA3 = 0.0
      NXTRA1 = 0
      NXTRA2 = 0

      NXTRA3 = 0
C
C
C *** GENERATE RADII AND VOLUMES ***
C
C
      RAD(1) = DR/4.0
      VOL(1) = DZ*PI*DR*DR/4.0
      DO 4 I=2,5
      FI = I
      RAD(I) = (FI-1.0)*DR
      VOL(I) = 2.0*PI*RAD(I)* DR * DZ
    4 CONTINUE
      RDS = RAD(5)
      LENG = 14.0 * DZ
      VOL(5) = VOL(5)/2.0
C
C
C *** FORMATS ***
C
C
  300 FORMAT(E10.3)
  301 FORMAT(I4)
  400 FORMAT(1H1, 50H +++++++ MPD ARC JET - LAGRANGE SOLUTION +++++++  /
     1      10X, 10H**INPUT**      /
     2      20X, 10H DT    =   , 1PE15.5/ 20X 10H MASSR =   , 1PE15.5/
     2      20X, 10H NAMPL =   , 1PE15.5/ 20X 10H EPSAMP=   , 1PE15.5/
     2      20X, 10H REFLC =   , 1PE15.5/ 20X 10H VOLTS =   , 1PE15.5/
     2      20X, 10H AMPSE =   , 1PE15.5/ 20X 10H AMPSI =   , 1PE15.5/
     2      20X, 10H NTSTEP=   , I4      / 20X, 10H OUTINC=   , I4   /
     2      20X, 10H OTTINC=   , I4      )
C
C
      LAST = 1
      RETURN
      END                                                              144
```

```
CMP04L3 SUBROUTINE FIELDS                VAL WATSON                          2696
      SUBROUTINE FIELDS
C
C
C
C***********************************************************************
C
C        THIS SUBROUTINE CALCULATES THE ELECTRIC POTENTIAL AND THE
C             MAGNETIC FIELD FROM THE PARTICLE DISTRIBUTIONS
C
C***********************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(2000), RE(2000), ZI(2000), ZE(2000), VRI(2000), VRE(2000),
     1     VZI(2000), VZE(2000)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(5,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)
      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1     NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST,L,M,K,R,Z,ZDIFF,ZAE,RP,ZP
      REAL CONST,C(5,6,17) ,CAA,CAC,CCA,CCC,CEC,CEA,QA,QC
C
C *************************************************************
C
C
C *** CALCULATION OF COEFFICIENTS FOR SOL OF POISS EQN BY DIRECT INTEG ***
C
C
      DATA  FIRST/0/
      IF(FIRST.GT.0) GO TO 10
      CALL CLOCK(CLOCK1)
      CALL POISS(C)
      CALL CLOCK(CLOCK2)
      CLOCK3 = CLOCK2 - CLOCK1
      WRITE(6,500) CLOCK3
  500 FORMAT(1H0,20H TIME FOR POISS IS  ,1PE10.3)
      CONST = DR*DR/(4.0*PI*EPSO*EPSAMP)
      DO 8 L=1,5
            DO 6 M=1,6
                  DO 4 K=1,16
                        C(L,M,K) = CONST*C(L,M,K)
    4             CONTINUE
    6       CONTINUE
    8 CONTINUE
      CAA = C(RA,RA+1,1)
      CAC = C(RA,RC,ZA+1)
      CCA = C(RC,RA+1,ZA)

      CCC = C(RC,RC,2)
      VOLA = 2.0*PI*4.5*DR*DZ*DR
      VOLC = VOL(RC)
C
      FIRST = 1
   10 CONTINUE
C
C
C *** ZERO THE NUM AND NUMV FOR EACH CELL ***
C
C
      DO 22 I=1,5
      DO 20 J=1,15
      NUMI(I,J) = 0.0
      NUMVI(I,J) = 0.0
      NUME(I,J) = 0.0
      NUMVE(I,J) = 0.0
      RHO(I,J) = 0.0
      JZ(I,J) = 0.0
   20 CONTINUE
   22 CONTINUE
C
C
C *** CALCULATE THE NUMBER OF PARTICLES IN EACH CELL ***
C
C
      DO 28 N=1,NIT
      I = RI(N) + 1.5
      J = ZI(N) + 1.5
      NUMI(I,J) = NUMI(I,J) + 1.0
      NUMVI(I,J) = NUMVI(I,J) + VZI(N)
      I = RI(N) + 1.0
      J = ZI(N) + 1.0
      FI = I
      FJ = J
      R1 = FI - RI(N)
      R2 = 1.0 - R1
      Z1 = FJ - ZI(N)
      Z2 = 1.0 - Z1
      RHO(I,J) = RHO(I,J) + R1 * Z1
      JZ(I,J) = JZ(I,J) + R1 * Z1 * VZI(N)
      IP = I + 1
      JP = J + 1
      RHO(IP,J) = RHO(IP,J) + R2 * Z1
      JZ(IP,J) = JZ(IP,J) + R2 * Z1 * VZI(N)
      RHO(I,JP) = RHO(I,JP) + R1 * Z2
       JZ(I,JP) =  JZ(I,JP) + R1 * Z2*VZI(N)
      RHO(IP,JP) = RHO(IP,JP) + R2 * Z2
       JZ(IP,JP) =  JZ(IP,JP) + R2 * Z2*VZI(N)
   28 CONTINUE
      DO 29 N=1,NET
      I = RE(N) + 1.5
      IF(I.LT.1) I = 1
      J = ZE(N) + 1.5
      IF(J.LT.1) J = 1
      NUME(I,J) = NUME(I,J) + 1.0
      NUMVE(I,J) = NUMVE(I,J) + VZE(N)
      I = RE(N) + 1.0
      J = ZE(N) + 1.0
      FI = I
      FJ = J
      R1 = FI - RE(N)
      R2 = 1.0 - R1
      Z1 = FJ - ZE(N)
      Z2 = 1.0 - Z1
      RHO(I,J) = RHO(I,J) - R1 * Z1
      JZ(I,J) = JZ(I,J) - R1 * Z1 * VZE(N)
      IP = I + 1
      JP = J + 1
      RHO(IP,J) = RHO(IP,J) - R2 * Z1
      JZ(IP,J) = JZ(IP,J) - R2 * Z1 * VZE(N)

      RHO(I,JP) = RHO(I,JP) - R1 * Z2
       JZ(I,JP) =  JZ(i,JP) - R1 * Z2*VZE(N)
      RHO(IP,JP) = RHO(IP,JP) - R2 * Z2
       JZ(IP,JP) =  JZ(IP,JP) - R2 * Z2*VZE(N)
   29 CONTINUE
C
C
C *** CALCULATION OF T. . CHARGE AND CURRENT DENSITIES ***
C
C
      DO 32 I=1,5
      DO 30 J=1,15
      RHO(I,J) = Q*(RHO(I,J)                  )/VOL(I) * NAMPL
      JZ(I,J) = Q*(JZ(I,J)                     )/VOL(I) * NAMPL
   30 CONTINUE
      JZ(I,1) = JZ(I,1)*2.0
   32 CONTINUE
C
C
C *** CALCULATION OF ELECTRIC POTENTIALS ***
CC *** SOLUTION OF POISSONS EQN BY DIRECT INTEGRATION ***
C
C
      IF(TSTEP.EQ.NTSTEP) CALL CLOCK (CLOCK1)
      DO 140 R=1,5
      DO 138 Z=1,15
      PHI(R,Z) = 0.0
      DO 136 ZP=1,15
      ZDIFF = IABS(ZP - Z) + 1
      DO 134 RP=1,5
      PHI(R,Z) = PHI(R,Z) + C(R,RP,ZDIFF)*RHO(RP,ZP)
  134 CONTINUE
  136 CONTINUE
      FZ = Z
      DIST = (30.0-FZ)*DZ
      PHI(R,Z) = PHI(R,Z) + EXTRA1*Q/(4.0*PI*EPSO*EPSAMP*DIST)
  138 CONTINUE
  140 CONTINUE
      IF(TSTEP.EQ.NTSTEP) CALL CLOCK(CLOCK2)
C
C
C *** MODIFICATION OF POTENTIALS ***
C
C
      QA = (VOLTS - (PHI(RA,ZA)-PHI(RC,ZC)) - QNET*(CAC-CCC)/VOLC)/
     1      ((CAA-CCA)/VOLA - (CAC-CCC)/VOLC)
      QC = QNET - QA
      RHOA = QA/VOLA
      RHOC = QC/VOLC
      DO 144 R=1,5
      DO 142 Z=1,15
      ZDIFF = IABS(ZA-Z) + 1
      PHI(R,Z) = PHI(R,Z) + C(R,RA+1,ZDIFF)*RHOA + C(R,RC,Z+1)*RHOC
  142 CONTINUE
  144 CONTINUE
C
      IF(TSTEP.NE.NTSTEP) GO TO 245
      CALL CLOCK(CLOCK3)
      CLKP1=CLOCK2-CLOCK1
      CLKP2=CLOCK3-CLOCK2
      WRITE(6,501) CLKP1,CLKP2
  501 FORMAT(1H0,30HTIME FOR POISS SOLUTION IS    ,1PE10.3/
     1           31H TIME FOR POISS ADJUSTMENT IS ,1PE10.3)
  245 CONTINUE
C
C
C *** CALCULATION OF ELECTRIC FIELDS ***
C
C
      DO 150 R=1,5
      DO 148 Z=1,15

      IF(R.EQ.1) ER(1,Z) =-(PHI(2,Z)-PHI(1,Z))*8.0/(15.0*DR)
      IF(R.EQ.5) ER(5,Z) =-(PHI(5,Z) - PHI(4,Z))/DR
      IF(R.EQ.1.OR.R.EQ.5) GO TO 146
      ER(R,Z) =-(PHI(R+1,Z) - PHI(R-1,Z))/(2.0*DR)
  146 CONTINUE
      IF(Z.EQ.1) EZ(R,1) =-(PHI(R,2) - PHI(R,1))/DZ
      IF(Z.EQ.15) EZ(R,15) =-(PHI(R,15) - PHI(R,14))/DZ
      IF(Z.EQ.1.OR.Z.EQ.15) GO TO 148
      EZ(R,Z) =-(PHI(R,Z+1) - PHI(R,Z-1))/(2.0*DZ)
  148 CONTINUE
  150 CONTINUE
C
C
C *** CALCULATION OF CURRENTS ***
C
C
      DO 155 R=2,5
  155 A(R,1) = -AMPSE
      DO 160 Z=2,15
      A(1,Z) = JZ(1,Z)*PI*RAD(1)**2
      A(2,Z) = PI*(DR/2.0)**2*JZ(1,7) + PI*RAD(2)*DR*JZ(2,Z)
      DO 158 R=3,5
      A(R,Z) = A(R-1,Z) + PI*DR*
     1      (RAD(R)*JZ(R,Z) + RAD(R-1)*JZ(R-1,Z))
  158 CONTINUE
  160 CONTINUE
C
C
C *** CALCULATION OF MAGNETIC FIELDS ***
C
C
      DO 170 Z=1,15
      DO 168 R=2,5
      B(R,Z) = A(R,Z)*MUZERO/(2.0*PI*RAD(R))
  168 CONTINUE

  170 CONTINUE
      DO 180 Z=2,14
      DO 178 R=2,5
      B(R,Z) = (B(R,Z-1) + B(R,Z) + B(R,Z+1))/3.0
  178 CONTINUE
  180 CONTINUE
C
      LAST = 1
      RETURN
      END
```

260

```
CMP0493 SUBROUTINE FOR POISSON EQUATION SOLUTION
      SUBROUTINE POISS(C)
C
C
C *** THIS SUBROUTINE CALCULATES THE POTENTIAL AT EACH MESH POIN1 DUE TO A
C         UNIT CHARGE IN EACH OTHER MESH CELL ***
C
C
C  CALCULATION OF DIMENSIONLESS COEFFICIENTS
      DIMENSION C(5,6,17)
      C(1,1,1) = 3.1416*0.64779
      NINC = 50
      FNINC = NINC
      DTH = 2.0*3.1416/FNINC
      DO 110 L = 1,5
           FL = L-1
           DO 100 N = 1,17
                FN = N-1
                IF(1-N) 12,10,10
   10           IF(1-L) 12,20,20
   12           C(L,1,N) = (3.1416/4.0)/SQRT(FL*FL + FN*FN)
   20           CONTINUE
                DO 90 M = 2,6
                     FM = M - 1
                     RC = 0.0
                     IF(M-5) 24,22,24
   22                FM = 3.75
   24                CONTINUE
                     IF(M-6) 28,26,28
   26                FM = 4.5
   28                CONTINUE
                     DO 80 J = 1,NINC
                          FJ = J - 1
                          THETA = FJ*DTH
                          IF(J-1) 30,30,60
   30                     IF(N-1) 40,40,60
   40                  IF(M-L) 60,50,60
   50                  CL = FM*FM*DTH*DTH/8.0
                       GO TO 70
   60                  CL = FM*DTH/SQRT((FM*COS(THETA)-FL)**2
     1                    +(FM*SIN(THETA))**2 + FN*FN)
   70                  RC = RC + CL
   80               CONTINUE
                    IF(M-5) 84,82,84
   82               RC = 0.5*RC
   84               CONTINUE
   90            C(L,M,N) = RC
  100         CONTINUE
  110      CONTINUE
           RETURN
           END
```

```
CMP04L4 SUBROUTINE TRAJ                VAL WATSON                            2696
      SUBROUTINE TRAJ
C
C
C
C***********************************************************************
C
C        THIS SUBROUTINE CALCULATES THE CHANGE IN PARTICLE VELOCITIES AND
C             POSITIONS FROM THE EQUATIONS OF MOTION USING THE CURRENT
C             VALUES OF ELECTRIC POTENTIAL AND MAGNETIC FIELD
C
C***********************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(2000), RE(2000), ZI(2000), ZE(2000), VRI(2000), VRE(2000),
     1     VZI(2000), VZE(2000)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(5,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)

      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1     NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST
      INTEGER INDXRC
      REAL RIO,ZIO,VRIO,VZIO,ERL,EZL,BL,ETA,THETA,ALPHA,THETA2,
     1   C1,C2,C3,C4
C **********************************************************
      DATA INDXRC/4/
      DATA NEXITC/0/
      NRECOM = 0
      DATA ATHRST/0.0/
      DATA AEXITF/0.0/
      THRUST = 0.0
      EXITFL = 0.0
C
      VRAND = SQRT(ABS(2.0*Q*VOLTS/MASSI)*0.1)
      IF(NIT.EQ.0) GO TO 300
      FZA = ZA
      ZANS = FZA - 1.5
      ZANE = FZA - 0.5
C
C
C *** ION TRAJECTORIES ***
C
C
      ETA = DT*Q/MASSI
      DO 100 N=1,NIT
```

```
C
C
C *** CHANGING FROM ARRAYS TO INDIVIDUAL LOCATIONS ***
C
C
      RIO = RI(N)
      ZIO = ZI(N)
      VRIO = VRI(N)
      VZIO = VZI(N)
C
C
C *** CALCULATION OF LOCAL FIELDS ***
C
C
      I = RIO + 1.0
      J = ZIO + 1.0
      FI = I-1
      FJ = J-1
      IF(I.EQ.5) GO TO 2
      IF(J.EQ.15) GO TO 4
      ERL = ER(I,J) +(RIO - FI) * (ER(I+1,J) - ER(I,J))
     1   + (ZIO - FJ) * (ER(I,J+1) - ER(I,J))
      EZL = EZ(I,J) + (RIO - FI) * (EZ(I+1,J) - EZ(I,J))
     1     + (ZIO - FJ) *(EZ(I,J+1) - EZ(I,J))
      BL = B(I,J) + (RIO - FI) * (B(I+1,J) - B(I,J))
     1    + (ZIO - FJ) * (B(I,J+1) - B(I,J))
      GO TO 6
    2 CONTINUE
      ERL = ER(I,J) + (ZIO - FJ) * (ER(I,J+1) - ER(I,J))
      EZL = EZ(I,J) + (ZIO - FJ) * (EZ(I,J+1) - EZ(I,J))
       BL =  B(I,J) + (ZIO - FJ) * ( B(I,J+1) -  B(I,J))
      GO TO 6
    4 CONTINUE
      ERL = ER(I,J) + (RIO - FI) * (ER(I+1,J) - ER(I,J))
      EZL = EZ(I,J) + (RIO - FI) * (EZ(I+1,J) - EZ(I,J))
       BL =  B(I,J) + (RIO - FI) * ( B(I+1,J) -  B(I,J))
    6 CONTINUE
C
C
C *** CALCULATION OF THE CONSTANTS ***
C
C
      THETA = ETA*BL/2.0
      THETA2 = THETA**2
      C3 = 1.0/(1.0+THETA2)
      C1 = (1.0-THETA2)*C3
      C4 = THETA*C3
      C2 = 2.0*C4
C
C
C *** TRAJECTORY EQUATIONS ***
C
C
      VZIN= C1*VZIO + C2*VRIO + C3*ETA*EZL + C4*ETA*ERL
      VRIN = -C2*VZIO + C1*VRIO - C4*ETA*EZL + C3*ETA*ERL
      RIN = RIO + DT*VRIN/DR
      ZIN = ZIO + DT*VZIN/DZ
C
C
C *** BOUNDARY CONDITIONS FOR THE PARTICLES ***
C
C
      IF(RIN.LT.4.0) GO TO 10
           RIN = 8.0  - RIN
           VRIN = -REFLC*VRIN
      VZIN = REFLC * VZIN
      GO TO 20
   10 CONTINUE
      IF(RIN.GT.0.0) GO TO 20
           RIN = -RIN
           VRIN = -VRIN
   20 CONTINUE

      IF(ZIN.LT.14.5) GO TO 30
           NLI = NLI + 1
           NDELI(NLI) = N
      NEXITC = NEXITC + 1
      THRUST = THRUST + VZI(N)
      EXITFL = EXITFL + 1.0
      GO TO 40
   30 CONTINUE
      IF(ZIN.GT.0.0) GO TO 40
      RADCE = RAD(RC)/DR + 0.25
      RADCS = RADCE-0.5
      IF(RIN.GT.RADCE.OR.RIN.LT.RADCS) GO TO 36
      INDXRC = INDXRC-1
      IF(INDXRC.GE.0) GO TO 36
      INDXRC = 4
      NLI = NLI+1
      NDELI(NLI) = N
      NRECOM = NRECOM+1
      GO TO 40
   36      ZIN = -ZIN
           VZIN = -REFLC*VZIN
      VRIN = REFLC * VRIN
   40 CONTINUE
C
C
C *** CHANGING BACK TO ARRAYS ***
C
C
      RI(N) = RIN
      ZI(N) = ZIN
      VRI(N) = VRIN
      VZI(N) = VZIN
  100 CONTINUE
C
```

```
C
C *** ELECTRON TRAJECTORIES ***
C
C
      ETA =-DT*Q/MASSE
      DO 200 N=1,NET
C
C
C *** CHANGING FROM ARRAYS TO INDIVIDUAL LOCATIONS ***
C
C
      RIO = RE(N)
      ZIO = ZE(N)
      VRIO = VRE(N)
      VZIO = VZE(N)
C
C
C *** CALCULATION OF LOCAL FIELDS ***
C
C
      I = RIO + 1.0
      J = ZIO + 1.0
      FI = I-1
      FJ = J-1
      IF(I.EQ.5) GO TO 102
      IF(J.EQ.15) GO TO 104
      ERL = ER(I,J) +(RIO - FI) * (ER(I+1,J) - ER(I,J))
     1   + (ZIO - FJ) * (ER(I,J+1) - ER(I,J))
      EZL = EZ(I,J) + (RIO - FI) * (EZ(I+1,J) - EZ(I,J))
     1     + (ZIO - FJ) *(EZ(I,J+1) - EZ(I,J))
      BL = B(I,J) + (RIO - FI) * (B(I+1,J) - B(I,J))
     1    + (ZIO - FJ) * (B(I,J+1) - B(I,J))
      GO TO 106
  102 CONTINUE
      ERL = ER(I,J) + (ZIO - FJ) * (ER(I,J+1) - ER(I,J))
      EZL = EZ(I,J) + (ZIO - FJ) * (EZ(I,J+1) - EZ(I,J))

       BL =  B(I,J) + (ZIO - FJ) * ( B(I,J+1) -  B(I,J))
      GO TO 106
  104 CONTINUE
      ERL = ER(I,J) + (RIO - FI) * (ER(I+1,J) - ER(I,J))
      EZL = EZ(I,J) + (RIO - FI) * (EZ(I+1,J) - EZ(I,J))
       BL =  B(I,J) + (RIO - FI) * ( B(I+1,J) -  B(I,J))
  106 CONTINUE
C
C
C *** CALCULATION OF THE CONSTANTS ***
C
C
      THETA = ETA*BL/2.0
      THETA2 = THETA**2
      C3 = 1.0/(1.0+THETA2)
      C1 = (1.0-THETA2)*C3
      C4 = THETA*C3
      C2 = 2.0*C4
C
C
C *** TRAJECTORY EQUATIONS ***
C
C
      VZIN= C1*VZIO + C2*VRIO + C3*ETA*EZL + C4*ETA*ERL
      VRIN = -C2*VZIO + C1*VRIO - C4*ETA*EZL + C3*ETA*ERL
      RIN = RIO + DT*VRIN/DR
      ZIN = ZIO + DT*VZIN/DZ
C
C
C *** BOUNDARY CONDITIONS FOR THE PARTICLES ***
C
C
      IF(ZIN.LT.ZANS.OR.ZIN.GT.ZANE.OR.RIN.LT.4.0) GO TO 108
      QNET = QNET - NAMPL*Q
      NLE = NLE + 1

      NDELE(NLE) = N
      GO TO 140
  108 IF(RIN.LT.4.0) GO TO 110
            RIN = 8.0  - RIN
            VRIN = -REFLC*VRIN
      VZIN = REFLC * VZIN
      GO TO 120
  110 CONTINUE
      IF(RIN.GT.0.0) GO TO 120
            RIN = -RIN
            VRIN = -VRIN
  120 CONTINUE
      IF(ZIN.LT.14.5) GO TO 130
            NLE = NLE + 1
            NDELE(NLE) = N
      NEXITC = NEXITC - 1
      GO TO 140
  130 CONTINUE
      IF(ZIN.GT.0.0) GO TO 140
            ZIN = -ZIN
            VZIN = -REFLC*VZIN
      VRIN = REFLC * VRIN
  140 CONTINUE
C
C
C *** CHANGING BACK TO ARRAYS ***
C
C
      RE(N) = RIN
      ZE(N) = ZIN
      VRE(N) = VRIN
      VZE(N) = VZIN
  200 CONTINUE
  300 CONTINUE
```

```
      THRUST = THRUST * NAMPL          * MASSI/DT

      ATHRST = (THRUST + 99. * ATHRST)/100.0
      EXITFL = EXITFL * NAMPL          * MASSI/DT
      AEXITF=(EXITFL + 99. * AEXITF)/100.0
      TTHRST = (AMPSI       *MASSI/Q)*SQRT(2.0*Q*VOLTS/MASSI)
      PTHRST =(THRUST/TTHRST)* 100.0
      PATHRS =(ATHRST/TTHRST)* 100.0
      TEXITF = AMPSI        *MASS:/Q
      PEXITF =(EXITFL/TEXITF) * 100.0
      PAEXIT =(AEXITF/TEXITF) * 100.0
      IF(OUTIND.GT.0) GO TO 500
      WRITE(6,400) TTHRST,PTHRST,PATHRS,TEXITF,PEXITF,PAEXIT
  400 FORMAT(1H1, 24HTHEORETICAL THRUST =      ,1PE15.4,10H NEWTONS    /
     1            24H ACTUAL THRUST =          ,2PE15.4,10H  PERCENT   /
     2            24H AVERAGE THRUST =         ,2PE15.4,10H  PERCENT   /
     3            25H THEORETICAL MASS FLOW =  ,1PE15.4,10H KG/SEC     /
     4            24H ACTUAL MASS FLOW =       ,2PE15.4,10H  PERCENT   /
     5            24H AVERAGE MASS FLOW =      ,2PE15.4,10H  PERCENT   )
  500 CONTINUE
      FNEXIT = NEXITC
      EXTRA1 = FNEXIT*NAMPL
      LAST = 1
      RETURN
      END
```

```
CMP04T5 SUBROUTINE SOURCE                    VAL WATSON                          2696
      SUBROUTINE SOURCE
C
C
C
C************************************************************
C
C           THIS SUBROUTINE PROVIDES FOR THE INJECTION OF PARTICLES INTO THE
C                ACCELERATOR BY VOLUME IONIZATION OR FROM THE ACCELERATOR
C                BOUNDARIES
C
C************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2    NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(2000), RE(2000), ZI(2000), ZE(2000), VRI.2000), VRE(2000),
     1     VZI(2000), VZE(2000)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(5,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)

      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1     NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST,NLOC,NSINDI,NSINDE
      REAL RSRCE(100),ZSRCE(100)
C **************************************************************
C
C
C *** ION SOURCE ***
      DATA NSINDI,NSINDE/1,1/
      DATA TLADD,TLADP,TLADDE,TLADPE/0.0,0.0,0.0,0.0/
      DATA FIRST/0/
      IF(FIRST.GT.0) GO TO 1
      FIRST = 1
C
C *** READ ARBITRARY LOCATIONS FOR IONIZATION ***
C
      READ(5,300) NLOC,( RSRCE(I),ZSRCE(I),I=1,NLOC)
      WRITE(6,302) (RSRCE(I),ZSRCE(I),I=1,NLOC)
      VRAND = SQRT(ABS(2.0*Q*VOLTS/MASSI)*0.1)
    1 CONTINUE
  300 FORMAT(I4/(2E10.3))
C *** DIFFERENCE BETWEEN THE TOTAL LOST AND THE TOTAL INPUT ***
      TLADD = TLADD + (AMPSI * DT)/(NAMPL * Q)
      NII = IFIX(TLADD) - IFIX(TLADP) + NRECOM
      TLADP = TLADD
C
C
```

```
      NDI = NII - NLI
C
C
C *** ASSIGNING PARTICLES TO THE N NUMBERS ***
C
C
C
C
C *** ASSIGNING NEW PARTICLES OF EACH RING ***
C
C
      IF(NII.LE.0) GO TO 20
      DO 6 NIND=1,NII
      IF(NLI.GT.0) GO TO 2
C
C *** SPACES NOT LEFT IN OLD ARRAY ***
C
C
      NIT = NIT+1
      IF(NIT.GT.2000) GO TO 400
      N = NIT
      GO TO 4
    2 CONTINUE
C
C
C *** SPACES LEFT IN OLD ARRAY ***
C
C
      N = NDELI(NLI)
      NLI = NLI-1
    4 CONTINUE
      RI(N) = RSRCE(NSINDI)
      ZI(N) = ZSRCE(NSINDI)
      NSINDI = NSINDI + 1
      IF(NSINDI.GT.NLOC) NSINDI = 1
      VRI(N)=(RANDOM(+1)-0.5)*VRAND

      VZI(N)=(RANDOM(+1)-0.5)*VRAND
    6 CONTINUE
   20 IF(NLI.LE.0) GO TO 30
C
C
C *** IF SPACES REMAIN AFTER ASSIGNING ALL NEW PARTICLES ***
C
C
      N = NDELI(NLI)
      NLI = NLI-1
      RI(N) = RI(NIT)
      ZI(N) = ZI(NIT)
      VRI(N) = VRI(NIT)
      VZI(N) = VZI(NIT)
      NIT = NIT-1
      GO TO 20
   30 CONTINUE
C
C
C *** ELECTRON SOURCE ***
      TLADDE = TLADDE + (AMPSE * DT)/(NAMPL * Q)
      NIE = IFIX(TLADDE) - IFIX(TLADPE)
      TLADPE = TLADDE
      NIE = NIE + NII - NRECOM
      FNREC = NRECOM
      QNET = QNET + NAMPL*Q*FNREC
C *** DIFFERENCE BETWEEN THE TOTAL LOST AND THE TOTAL INPUT ***
C
C
      NDE = NIE - NLE
C
C
C *** ASSIGNING PARTICLES TO THE N NUMBERS ***
C
```

```
C
C
C
C *** ASSIGNING NEW PARTICLES OF EACH RING ***
C
C
      IF(NIE.LE.0) GO TO 120
      DO 106 NIND = 1,NIE
      IF(NLE.GT.0) GO TO 102
C
C *** SPACES NOT LEFT IN OLD ARRAY ***
C
C
      NET = NET+1
      IF(NET.GT.2000) GO TO 400
      N = NET
      GO TO 104
  102 CONTINUE
C
C
C *** SPACES LEFT IN OLD ARRAY ***
C
C
      N = NDELE(NLE)
      NLE = NLE-1
  104 CONTINUE
      IF(NIND.GT.NII) GO TO 105
      RE(N) = RSRCE(NSINDE)
      ZE(N) = ZSRCE(NSINDE)
      NSINDE = NSINDE + 1
      IF(NSINDE.GT.NLOC) NSINDE = 1
      VRE(N)=(RANDOM(+1)-0.5)*VRAND
      VZE(N)=(RANDOM(+1)-0.5)*VRAND
      GO TO 106
  105 RE(N) = RANDOM(+1)/2.0
```

```
      ZE(N) = 0.001
      VRE(N)=(RANDOM(+1)-0.5)*VRAND
      VZE(N)= RANDOM(+1) * VRAND
      QNET = QNET + NAMPL*Q
  106 CONTINUE
  120 IF(NLE.LE.0) GO TO 130
C
C
C *** IF SPACES REMAIN AFTER ASSIGNING ALL NEW PARTICLES ***
C
C
      N = NDELE(NLE)
      NLE = NLE-1
      RE(N) = RE(NET)
      ZE(N) = ZE(NET)
      VRE(N) = VRE(NET)
      VZE(N) = VZE(NET)
      NET = NET-1
      GO TO 120
  130 CONTINUE
      GO TO 500
  400 NEXCD = 1
  500 CONTINUE
  302 FORMAT(1H0,20H SOURCE LOCATIONS   /(1P2E10.3))
      LAST = 1
      RETURN
      END
```

```
CMP04T6 SUBROUTINE OUTPT                    VAL WATSON                    2696
      SUBROUTINE OUTPT
C
C
C
C*********************************************************************
C
C         THIS SUBROUTINE PRINTS THE LOCAL VALUES OF THE FIELDS,
C              DENSITIES, AND CURRENTS
C
C*********************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(2000), RE(2000), ZI(2000), ZE(2000), VRI(2000), VRE(2000),
     1     VZI(2000), VZE(2000)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(5,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400). NDELE(400)
      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1       NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST
C ************************************************************
C
      IF(TSTEP.EQ.0.AND.T.GT.0.0) GO TO 200
      NFILE = NFILE + 1
      WRITE(8)  T,PHI,B,NUMI,NUME,NUMVI,NUMVE,NIT,(RI(I),ZI(I),VRI(I),VZ
     1I(I),I=1,NIT),NET,(RE(I),ZE(I).VRE(I),VZE(I),I=1,NET),NFRAME,QNET
      END FILE 8
  200 WRITE(6,300) T,NFILE,NFRAME
      WRITE(6,310)
      WRITE(6,320) ((PHI(I,J), I=1,5),J=1,15)
      WRITE(6,330)
      WRITE(6,320) ((    B(I,J),I=1,5),J=1,15)
      WRITE(6,340)
      WRITE(6,320) ((NUMI(I,J),I=1,5),J=1,15)
      WRITE(6,350)
      WRITE(6,320) ((NUME(I,J),I=1,5),J=1,15)
      WRITE(6,360)
      WRITE(6,320) ((NUMVI(I,J),I=1,5),J=1,15)
      WRITE(6,370)
      WRITE(6,320) ((NUMVE(I,J),I=1,5),J=1,15)
  300 FORMAT(1H1,  10H TIME =    , 1PE15.5, 5X, 8H FILE NO  ,I4,
     1     5X, 9H FRAME NO   , I4)
  310 FORMAT(1H0, 20H *** POTENTIALS ***            )
  320 FORMAT(1P5E13.5)
  330 FORMAT(1H0, 30H *** MAGNETIC FIELDS ***                   )
```

```
340 FORMAT(1H1, 40H *** NUMBER OF IONS IN EACH CELL ***         )

350 FORMAT(1H0, 40H *** NUMBER OF ELECTRONS PER CELL ***        )
360 FORMAT(1H1, 40H *** MASS FLOW OF IONS ***                   )
370 FORMAT(1H0, 40H *** MASS FLOW OF ELECTRONS ***              )
    LAST = 1
    RETURN
    END
```

```
CMP04T7 SUBROUTINE OTTPT              VAL WATSON                          2696
      SUBROUTINE OTTPT
C
C
C
C*********************************************************************
C
C        THIS SUBROUTINE CONVERTS THE VALUES FOR THE PARTICLE POSITIONS
C             AND THE FIELDS TO 12 BIT INTEGERS AND STORES 3 PER WORD
C             ON MAGNETIC TAPE FOR FUTURE PLOTTING
C
C*********************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(2000), RE(2000), ZI(2000), ZE(2000), VRI(2000), VRE(2000),
     1     VZI(2000), VZE(2000)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(5,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)

      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1     NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST
      INTEGER IARRAY(770)
C **************************************************************
C
      IF(TSTEP.GT.1) GO TO 100
      DO 60 IA = 1,380
      IARRAY(IA) = 0
   60 CONTINUE
      READ(5,500) TAMP,PHIAMP,BAMP,RAMP,ZAMP
      WRITE(6,540) TAMP,PHIAMP,BAMP,RAMP,ZAMP
  100 CONTINUE
      IF(TSTEP.EQ.1) GO TO 600
      IARRAY(1) = 0
      IARRAY(2) = T*TAMP
      IARRAY(3) = NIT
      IARRAY(4) = NET
      IA = 4
C
C *** CONVERT INFORMATION TO BE PLOTTED INTO 12 BIT INTEGERS ***
C
      DO 140 J = 1,15
      DO 120 I = 1,5
           IA = IA + 1
           IARRAY(IA) = PHI(I,J)*PHIAMP + 200.0
  120 CONTINUE
  140 CONTINUE

      DO 200 J = 1,15
      DO 180 I = 1,5
             IA = IA + 1
             IARRAY(IA) = B(I,J)*BAMP + 200.0
  180 CONTINUE
  200 CONTINUE
      CALL   FOLD(IARRAY(2),153)
C
C         SUBROUTINE FOLD PACKS 3 12 BIT INTEGERS INTO EACH 7094 BIT WORD ***
C
      WRITE(10) (IARRAY(I),I=1,53)
      DO 240 I=1,770
  240  IARRAY(I) = 0
      IA = 1
      DO 300 I=1,NIT
             IA = IA + 1
             IARRAY(IA) = RI(I)*RAMP
             IA = IA + 1
             IARRAY(IA) = ZI(I)*ZAMP
             IF(IA.LT.763) GO TO 300
             CALL   FOLD(IARRAY(2),762)
             WRITE(10) (IARRAY(J),J=1,255)
             IA = 1
  300 CONTINUE
      IF(IA.EQ.1) GO TO 320
      IARL = 1000 - (3000-IA+1)/3
      IARL3 = IARL*3
      IARLP = IARL+1
      IA = IA+1
      IARRAY(IA) = 0
      IA = IA+1
      IARRAY(IA) = 0
      CALL FOLD(IARRAY(2),IARL3)
      WRITE(10) (IARRAY(I),I=1,IARLP)
```

```
320 CONTINUE

    IA = 1
    DO 400 I=1,NET
          IA = IA + 1
          IARRAY(IA) = RE(I)*RAMP
          IA = IA + 1
          IARRAY(IA) = ZE(I)*ZAMP
          IF(IA.LT.763) GO TO 400
          CALL   FOLD(IARRAY(2),762)
          WRITE(10) (IARRAY(J),J=1,255)
          IA = 1
400 CONTINUE
    IF(IA.EQ.1) GO TO 420
    IARL = 1000 - (3000-IA+1)/3
    IARL3 = IARL*3
    IARLP = IARL+1
    IA = IA+1
    IARRAY(IA) = 0
    IA = IA+1
    IARRAY(IA) = 0
    CALL FOLD(IARRAY(2),IARL3)
    WRITE(10) (IARRAY(I),I=1,IARLP)
420 CONTINUE
    NFRAME = NFRAME + 1
    LAST = 1
500 FORMAT(5E10.3)
540 FORMAT(2H , 9HTAMP    =    ,E10.3/ 10H PHIAMP =     ,E10.3/
   1           10H BAMP    =    ,E10.3/ 10H RAMP   =     ,E10.3/
   2           10H ZAMP    =    ,E10.3)
600 RETURN
    END
```

```
CMP04D1 *** DIRECT ELECTROMAGNETIC ACCELERATOR *** VAL WATSON               2696
C
C
C
C***********************************************************************
C
C        MAIN PROGRAM FOR THE COMPUTER SIMULATION OF CHENGS
C                  AXISYMMETRIC PLASMA ACCELERATOR
C
C***********************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(1800), RE(1800), ZI(1800), ZE(1800), VRI(1800), VRE(1800),
     1     VZI(1800), VZE(1800)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(6,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)
      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1      NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NExCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST
C ********************************************************
C
C
      FIRST = 1
C
C
C *** INITIAL VALUES ***
C
C
      CALL INITL
C        SUBROUTINE INITL INITIALIZES ALL OF THE COMMON VARIABLES
C
      CLOCK4 = 0.0
C
C
C *** TIME LOOP ***
C
C
      DO 200 TSTEP=1,NTSTEP
      CALL FIELDS
C        SUBROUTINE FIELDS EVALUATES THE ELECTRIC AND MAGNETIC FIELDS
C        FROM THE PARTICLE DISTRIBUTIONS
C
C
C *** OUTPUT ON TAPE FOR FUTURE PLOTTING ***
C
      IF(OTTINC.EQ.0) GO TO 80
      IF(TSTEP.EQ.NTSTEP) GO TO 40

      IF(OTTIND.GT.0) GO TO 80
      OTTIND = OTTINC
   40 CALL OTTPT
C        SUBROUTINE OTTPT STORES THE PARTICLE POSITIONS AND THE FIELDS ON
C        MAGNETIC TAPE
C
   80 OTTIND = OTTIND - 1
      IF(OUTIND.GT.0) GO TO 100
      OUTIND = OUTINC
C
C *** WRITTEN OUTPUT ***
C
      CALL OUTPT
C        SUBROUTINE OUTPUT PRINTS THE VALUES FOR THE LOCAL FIELDS,
C        DENSITIES, AND CURRENTS
C
      CLOCKP = CLOCK4
      CALL CLOCK(CLOCK4)
      CLOCKS=CLOCK4-CLOCKP
      WRITE(6,700) CLOCKS
  700 FORMAT(1HO,20H TIME FOR OUTPT IS  ,1PE10.3)
  100 OUTIND = OUTIND - 1
C
C
      T = T + DT
      DT = DT*EX
      IF(TSTEP.GT.1) CALL TRAJ
C        SUBROUTINE TRAJ CALCULATES THE PARTICLE VELOCITIES AND POSTIONS
C        AT EACH TIME STEP
C
      CALL SOURCE
C        SUBROUTINE SOURCE PROVIDES FOR INJECTION OF PARTICLES INTO THE
C        ACCELERATOR EITHER BY VOLUME IONIZATION OR FROM THE BOUNDARIES
C
```

```
      IF(NEXCD.GT.0) GO TO 400
  200 CONTINUE
      GO TO 500
  400 WRITE(6,600)
  500 CONTINUE
  600 FORMAT(10X,30H EXCEEDED 2000 PARTICLES       )
      LAST = 1
      RETURN
      END
```

```
CMPO4D2 SUBROUTINE INITL                     VAL WATSON                             2696
      SUBROUTINE INITL
C
C
C
C***********************************************************************
C
C        THIS SUBROUTINE INITIALIZES THE COMMON VARIABLES
C
C***********************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(1800), RE(1800), ZI(1800), ZE(1800), VRI(1800), VRE(1800),
     1     VZI(1800), VZE(1800)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(6,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)
      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1     NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NExCD

      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST
C ***********************************************************
      CALL LOCATE(1,8)
      CALL LOCATE(1,10)
C
C
C *** INPUT DATA ***
C
C
      READ(5,300) DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,AMPSE,AMPSI
      READ(5,301) NTSTEP,OUTINC,OTTINC
      WRITE(6,400) DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,AMPSE,AMPSI
     1     ,NTSTEP,OUTINC,OTTINC
C
C
C *** SET FIXED CONSTANTS ***
C
C
      EX = 1.0
      DR = 0.01
      DZ = 0.01
      PI = 3.14159
      Q = 1.6E-19
      QNET = 3.0E-9
      MUZERO = 4.0*PI*1.0E-7
      EPSO = 8.85E-12
      MASSI = 1.67E-27
      MASSE = MASSI/MASSR
C
C
C ***  SET INTERNAL VARIABLES ***
C
C
      T = 0.0
      RA=5
      ZA=5
      RC = 1
      ZC = 1
      OUTIND = 0
      OTTIND = 0
      NFILE = 0
      NFRAME = 0
      NIT = 0
      NLI = 0
      N = 0
      NET = 0
      NLE = 0
      NEXCD = 0
      NRECOM = 0
      RE(1) = 0.0
      RI(1) = 0.0
      ZE(1) = 0.0
      ZI(1) = 0.0
      VRE(1) = 0.0
      VRI(1) = 0.0
      VZE(1) = 0.0
      VZI(1) = 0.0
      DO 2 M=1,400
      NDELI(M) = 0
      NDELE(M) = 0
    2 CONTINUE
      EXTRA1 = 0.0
      EXTRA2 = 0.0

      EXTRA3 = 0.0
      NXTRA1 = 0
      NXTRA2 = 0
      NXTRA3 = 0
C
C
C *** GENERATE RADII AND VOLUMES ***
C
C
      RAD(1) = DR/4.0
      VOL(1) = DZ*PI*DR*DR/4.0
      DO 4 I=2,5
      FI = I
      RAD(I) = (FI-1.0)*DR
      VOL(I) = 2.0*PI*RAD(I)* DR * DZ
    4 CONTINUE
      RDS = RAD(5)
      LENG = 14.0 * DZ
      VOL(5) = VOL(5)/2.0
C
C
C *** FORMATS ***
C
C
  300 FORMAT(E10.3)
  301 FORMAT(I4)
  400 FORMAT(1H1, 50H +++++++ MPD ARC JET - LAGRANGE SOLUTION +++++++  /
     1      10X, 10H**INPUT**       /
     2      20X, 10H DT     =  , 1PE15.5/ 20X 10H MASSR =  , 1PE15.5/
     2      20X, 10H NAMPL  =  , 1PE15.5/ 20X 10H EPSAMP=  , 1PE15.5/
     2      20X, 10H REFLC  =  , 1PE15.5/ 20X 10H VOLTS =  , 1PE15.5/
     2      20X, 10H AMPSE  =  , 1PE15.5/ 20X 10H AMPSI =  , 1PE15.5/
     2      20X, 10H NTSTEP=   , I4     / 20X, 10H OUTINC=  , I4   /
     2      20X, 10H OTTINC=   , I4     )
```

```
C
C
      LAST = 1
      RETURN
      END
```

```
CMP04DB SUBROUTINE INITL                    VAL WATSON                       2696
      SUBROUTINE INITL
C
C
C
C*********************************************************************
C
C           THIS SUBROUTINE INITIALIZES THE COMMON VARIABLES FOR A
C                     CONTINUATION RUN
C
C*********************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(1800), RE(1800), ZI(1800), ZE(1800), VRI(1800), VRE(1800),
     1     VZI(1800), VZE(1800)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(6,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)
      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1     NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST
C ****************************************************************
C
C
C *** INPUT DATA ***
C
C
      CALL LOCATE(1,10)
      READ(5,301) NFILE
      CALL LOCATE(NFILE,8)
      READ(5,300) DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,AMPSE,AMPSI
      READ(5,301) NTSTEP,OUTINC,OTTINC
      WRITE(6,400) DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,AMPSE,AMPSI
     1    ,NTSTEP,OUTINC,OTTINC
C
C
C *** SET FIXED CONSTANTS ***
C
C
      EX = 1.0
      DR = 0.01
      DZ = 0.01
      PI = 3.14159
      Q = 1.6E-19
      MUZERO = 4.0*PI*1.0E-7
      EPSO = 8.85E-12
      MASSI = 1.67E-27
      MASSE = MASSI/MASSR
C
C
C ***  SET INTERNAL VARIABLES ***
C
C
      READ(8)   T,PHI,B,NUMI,NUME,NUMVI,NUMVE,NIT,(RI(I),ZI(I),VRI(I),VZ
     1I(I),I=1,NIT),NET,(RE(I),ZE(I),VRE(I),VZE(I),I=1,NET),NFRAME,QNET
      CALL SKIP(1,8)
      RA=5
      ZA=5
      RC = 1
      ZC = 1
      OUTIND = 0
      OTTIND = 0
      NLI = 0
      N = 0
      NLE = 0
      NEXCD = 0
      NRECOM = 0
      RE(1) = 0.0
      RI(1) = 0.0
      ZE(1) = 0.0
      ZI(1) = 0.0
      VRE(1) = 0.0
      VRI(1) = 0.0
      VZE(1) = 0.0
      VZI(1) = 0.0
      DO 2 M=1,400
      NDELI(M) = 0
      NDELE(M) = 0
    2 CONTINUE
      EXTRA1 = 0.0
      EXTRA2 = 0.0
      EXTRA3 = 0.0
    NXTRA1 = 0

    NXTRA2 = 0
    NXTRA3 = 0
C
C
C *** GENERATE RADII AND VOLUMES ***
C
C
    RAD(1) = DR/4.0
    VOL(1) = DZ*PI*DR*DR/4.0
    DO 4 I=2,5
    FI = I
    RAD(I) = (FI-1.0)*DR
    VOL(I) = 2.0*PI*RAD(I)* DR * DZ
  4 CONTINUE
    RDS = RAD(5)
    LENG = 14.0 * DZ
    VOL(5) = VOL(5)/2.0
C
C
C *** FORMATS ***
C
C
 300 FORMAT(E10.3)
 301 FORMAT(I4)
 400 FORMAT(1H1, 50H +++++++ MPD ARC JET - LAGRANGE SOLUTION +++++++  /
    1    10X, 10H**INPUT**      /
    2    20X, 10H DT     =  , 1PE15.5/ 20X 10H MASSR =   , 1PE15.5/
    2    20X, 10H NAMPL  =  , 1PE15.5/ 20X 10H EPSAMP=   , 1PE15.5/
    2    20X, 10H REFLC  =  , 1PE15.5/ 20X 10H VOLTS =   , 1PE15.5/
    2    20X, 10H AMPSE  =  , 1PE15.5/ 20X 10H AMPSI =   , 1PE15.5/
    2    20X, 10H NTSTEP=   , I4     / 20X, 10H OUTINC=   , I4   /
    2    20X, 10H OTTINC=   , I4     )
C
C
```

```
      LAST = 1

      RETURN
      END                                                                145
```

```
CMP04D3 SUBROUTINE FIELDS                   VAL WATSON                        2696
      SUBROUTINE FIELDS
C
C
C
C***********************************************************************
C
C        THIS SUBROUTINE CALCULATES THE ELECTRIC POTENTIAL AND THE
C             MAGNETIC FIELD FROM THE PARTICLE DISTRIBUTIONS
C
C***********************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG.AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(1800), RE(1800), ZI(1800), ZE(1800), VRI(1800), VRE(1800),
     1     VZI(1800), VZE(1800)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(6,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZE$O,EPSC
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)
      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1     NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST,L,M,K,R,Z,ZDIFF,ZAE,RP,ZP
      REAL CONST,C(5,6,17) ,CAA,CAC,CCA,CCC,CEC,CEA,QA,QC
      REAL CM(16,16)
      REAL VECT(16)
      REAL PHIO(5,15)
      REAL BO(5,15)
C ***********************************************************
C
C
C *** CALCULATION OF COEFFICIENTS FOR SOL OF POISS EQN BY DIRECT  INTEG ***
C
C
      DATA  FIRST/0/
      IF(FIRST.GT.0) GO TO 10
      FIRST = 1
      CALL CLOCK(CLOCK1)
      CALL POISS(C)
      CALL CLOCK(CLOCK2)
      CLOCK3 = CLOCK2 - CLOCK1
      WRITE(6,500) CLOCK3
  500 FORMAT(1HC,20H TIME FOR POISS IS  ,1PE10.3)
      CONST = DR*DR/(4.0*PI*EPSO*EPSAMP)
      DO 8 L=1,5
           DO 6 M=1,6
                DO 4 K=1,16
                     C(L,M,K) = CONST*C(L,M,K)
```

```
    4             CONTINUE

    6         CONTINUE
    8 CONTINUE
      VOLA = 2.0*PI*4.5*DR*DZ*DR
      VOLC = VOL(RC)
      ENDCC = 1.0/(4.0 * PI * EPSO * DZ)
      CALL MAT(C,VOLA,VOLC,CM)
C        SUBROUTINE MAT CALCULATES THE MATRIX REQUIDED FOR CALCULATION OF
C        THE ELECTRODE SURFACE CHARGES THAT SATISFY THE CONDITIONS FOR
C        CONSTANT POTENTIAL ON EACH ELECTRODE, CONSTANT POTENTIAL BETWEEN
C        ELECTRODES, AND CONSERVATION OF CHARGE
      DO 4000 J=1,15
 4000 B(1,J) = 0.0
   10 CONTINUE
C
C
C *** ZERO THE NUM AND NUMV FOR EACH CELL ***
C
C
      DO 22 I=1,5
      DO 20 J=1,15
      NUMI(I,J) = 0.0
      NUMVI(I,J) = 0.0
      NUME(I,J) = 0.0
      NUMVE(I,J) = 0.0
   20 CONTINUE
   22 CONTINUE
C
C
C *** CALCULATE THE NUMBER OF PARTICLES IN EACH CELL ***
C
C
      DO 28 N=1,NIT
      I = RI(N) + 1.5
      J = ZI(N) + 1.5
      NUMI(I,J) = NUMI(I,J) + 1.0

      NUMVI(I,J) = NUMVI(I,J) + VZI(N)
      I = RI(N) + 1.0
      J = ZI(N) + 1.0
      FI = I
      FJ = J
      R1 = FI - RI(N)
      R2 = 1.0 - R1
      Z1 = FJ - ZI(N)
      Z2 = 1.0 - Z1
      JZ(I,J) = JZ(I,J) + R1 * Z1 * VZI(N)
      IP = I + 1
      JP = J + 1
      RHO(IP,J) = RHO(IP,J) + R2 * Z1
      JZ(IP,J) = JZ(IP,J) + R2 * Z1 * VZI(N)
      RHO(I,JP) = RHO(I,JP) + R1 * Z2
       JZ(I,JP) =  JZ(I,JP) + R1 * Z2*VZI(N)
      RHO(IP,JP) = RHO(IP,JP) + R2 * Z2
       JZ(IP,JP) =  JZ(IP,JP) + R2 * Z2*VZI(N)
   28 CONTINUE
      RHO(I,J) = RHO(I,J) + R1 * Z1
      DO 29 N=1,NET
      I = RE(N) + 1.5
      J = ZE(N) + 1.5
      NUME(I,J) = NUME(I,J) + 1.0
      NUMVE(I,J) = NUMVE(I,J) + VZE(N)
      I = RE(N) + 1.0
      J = ZE(N) + 1.0
      FI = I
      FJ = J
      R1 = FI - RE(N)
      R2 = 1.0 - R1
      Z1 = FJ - ZE(N)
      Z2 = 1.0 - Z1
      RHO(I,J) = RHO(I,J) - R1 * Z1
      JZ(I,J) = JZ(I,J) - R1 * Z1 * VZE(N)
      IP = I + 1
      JP = J + 1
      RHO(IP,J) = RHO(IP,J) - R2 * Z1
      JZ(IP,J) = JZ(IP,J) - R2 * Z1 * VZE(N)
      RHO(I,JP) = RHO(I,JP) - R1 * Z2
       JZ(I,JP) =  JZ(I,JP) - R1 * Z2*VZE(N)
      RHO(IP,JP) = RHO(IP,JP) - R2 * Z2
       JZ(IP,JP) =  JZ(IP,JP) - R2 * Z2*VZE(N)
   29 CONTINUE
C
C
C *** CALCULATION OF THE CHARGE AND CURRENT DENSITIES ***
C
C
      DO 32 I=1,5
      DO 30 J=1,15
      RHO(I,J) = Q*(RHO(I,J)                  )/VOL(I) * NAMPL
      JZ(I,J) = Q*(JZ(I,J)                     )/VOL(I) * NAMPL
   30 CONTINUE
      JZ(I,1) = JZ(I,1)*2.0
   32 CONTINUE
C
C
C *** CALCULATION OF ELECTRIC POTENTIALS ***
CC *** SOLUTION OF POISSONS EQN BY DIRECT INTEGRATION ***
C
C
      IF(TSTEP.EQ.NTSTEP) CALL CLOCK (CLOCK1)
      DO 140 R=1,5
      DO 138 Z=1,15
      PHI(R,Z) = 0.0
      DO 136 ZP=1,15
      ZDIFF = IABS(ZP - Z) + 1
      DO 134 RP=1,5
      PHI(R,Z) = PHI(R,Z) + C(R,RP,ZDIFF)*RHO(RP,ZP)
```

```
  134 CONTINUE

  136 CONTINUE
      ENDCD = (24  - Z)
      PHI(R,Z) = PHI(R,Z) + ENDCC * EXTRA1/ENDCD
  138 CONTINUE
  140 CONTINUE
      IF(TSTEP.EQ.NTSTEP) CALL CLOCK(CLOCK2)
      DO 1000 J = 1,8
 1000 VECT(J) = VOLTS - PHI(5,J) + PHI(1,J)
      DO 1010 J = 9,15
 1010 VECT(J) = -PHI(5,J-8) + PHI(5,J-7)
      VECT(16) = QNET * 1.0E+10
      DO 1030 J = 1,8
      RHO(1,J) = 0.0
      RHO(6,J) = 0.0
      DO 1020 I = 1,16
      RHO(6,J) = RHO(6,J) + CM(J,I) * VECT(I)
      RHO(1,J) = RHO(1,J) + CM(J+8,I) * VECT(I)
 1020 CONTINUE
 1030 CONTINUE
C
C
C *** MODIFICATION OF POTENTIALS ***
C
C
      DO 144 R = 1,5
      DO 143 Z = 1,15
      DO 142 ZP = 1,8
      ZDIFF = IABS(ZP-Z)+1
      DO 141 RP = 1,6,5
      PHI(R,Z) = PHI(R,Z) + C(R,RP,ZDIFF) * RHO(RP,ZP)
  141 CONTINUE
  142 CONTINUE
  143 CONTINUE
  144 CONTINUE
```

```
      IF(TSTEP.NE.NTSTEP) GO TO 245

      CALL CLOCK(CLOCK3)
      CLKP1=CLOCK2-CLOCK1
      CLKP2=CLOCK3-CLOCK2
      WRITE(6,501) CLKP1,CLKP2
  501 FORMAT(1H0,30HTIME FOR POISS SOLUTION IS    ,1PE10.3/
     1            31H TIME FOR POISS ADJUSTMENT IS ,1PE10.3)
  245 CONTINUE
C
C
C *** CALCULATION OF ELECTRIC FIELDS ***
C
C
      DO 150 R=1,5
      DO 148 Z=1,15
      IF(R.EQ.1) ER(1,Z) =-(PHI(2,Z)-PHI(1,Z))*2.0/DR
      IF(R.EQ.5) ER(5,Z) =-(PHI(5,Z) - PHI(4,Z))/DR
      IF(R.EQ.1.OR.R.EQ.5) GO TO 146
      ER(R,Z) =-(PHI(R+1,Z) - PHI(R-1,Z))/(2.0*DR)
  146 CONTINUE
      IF(Z.EQ.1) EZ(R,1) =-(PHI(R,2) - PHI(R,1))/DZ
      IF(Z.EQ.15) EZ(R,15) =-(PHI(R,15) - PHI(R,14))/DZ
      IF(Z.EQ.1.OR.Z.EQ.15) GO TO 148
      EZ(R,Z) =-(PHI(R,Z+1) - PHI(R,Z-1))/(2.0*DZ)
  148 CONTINUE
  150 CONTINUE
C
C
C *** CALCULATION OF CURRENTS ***
C
C
      DO 155 R=2,5
  155 A(R,1) = -AMPSE
      DO 160 Z=2,8
      FRZ = FLOAT(Z-1)/7.0
      A(1,Z) =-(1.0 - FRZ) * AMPSE

      A(2,Z) =-(1.0-FRZ) * AMPSE      + PI*RAD(2)*DR*JZ(2,Z)
      DO 158 R=3,5
      A(R,Z) = A(R-1,Z) + PI*DR*
     1       (RAD(R)*JZ(R,Z) + RAD(R-1)*JZ(R-1,Z))
  158 CONTINUE
  160 CONTINUE
      DO 164 Z = 9,15
      A(1,Z) = JZ(1,Z)*PI*RAD(1)**2
      A(2,Z) = PI*(DR/2.0)**2*JZ(1,Z) + PI*RAD(2)*DR*JZ(2,Z)
      DO 162 R=3,5
      A(R,Z) = A(R-1,Z) + PI*DR*
     1       (RAD(R)*JZ(R,Z) + RAD(R-1)*JZ(R-1,Z))
  162 CONTINUE
  164 CONTINUE
C
C
C *** CALCULATION OF MAGNETIC FIELDS ***
C
C
      DO 170 Z=1,15
      DO 168 R=2,5
      B(R,Z) = A(R,Z)*MUZERO/(2.0*PI*RAD(R))
  168 CONTINUE
  170 CONTINUE
      DO 180 Z=2,14
      DO 178 R=1,5
      B(R,Z) = (B(R,Z-1) + B(R,Z) + B(R,Z+1))/3.0
  178 CONTINUE
  180 CONTINUE
      LAST = 1
      RETURN
      END
```

```
CMP04D8     SUBROUTINE MAT
      SUBROUTINE MAT(C,VOLA,VOLC,CM)
C
C *** THIS SUBROUTINE CALCULATES THE MATRIX REQUIRED FOR CALCULATION OF
C        THE ELECTRODE SURFACE CHARGES THAT SATISFY THE CONDITIONS FOR
C        CONSTANT POTENTIAL ON EACH ELECTRODE, CONSTANT POTENTIAL BETWEEN
C        ELECTRODES, AND CONSERVATION OF CHARGE ***
C
      FIRST = 1.0
      REAL C(5,6,17) ,CM(16,16)
      DO 20 J = 1,8
         JP = J + 8
         CM(1,J) = C(5,6,J) - C(1,6,J)
         CM(1,JP)= C(5,1,J) - C(1,1,J)
   20 CONTINUE
         DO 40 I = 2,8
      CM(I,1) = CM(1,I)
      CM(I,9) = CM(1,I+8)
         DO 30 J = 2,8
            JP = J + 8
            CM(I,J) = CM(I-1,J-1)
            CM(I,JP)= CM(I-1,JP-1)
   30       CONTINUE
   40       CONTINUE
         DO 50 J = 2,8
            JP = J + 8
            CM(9,J)  = C(5,6,J) - C(5,6,J-1)
            CM(9,JP) = C(5,1,J) - C(5,1,J-1)
   50       CONTINUE
            CM(9,1) = -CM(9,2)
            CM(9,9) = -CM(9,10)
      DO 70 I = 10,15
            CM(I,1) = -CM(9,I-7)
            CM(I,9) = -CM(9,I+1)
          DO 60 J = 2,8
            JP = J + 8
```

```
               CM(I,J) = CM(I-1,J-1)

               CM(I,JP)= CM(I-1,JP-1)
   60 CONTINUE
   70 CONTINUE
      DO 90 J = 1,8
               JP = J + 8
               CM(16,J) = VOLA * 1.0E10
               CM(16,JP)= VOLC * 1.0E10
   90 CONTINUE
  100 CONTINUE
      DO 200 I = 1,16
      DO 150 J = 1,16
            CM(I,J) = CM(I,J) * 1.0E-6
  150 CONTINUE
  200 CONTINUE
      CALL MINV(CM,16,B,0,DET)
      DO 300 I = 1,16
      DO 250 J = 1,16
            CM(I,J) = CM(I,J) * 1.0E-6
  250 CONTINUE
  300 CONTINUE
      LAST = 1
      RETURN
      END
```

```
CMP04M4 SUBROUTINE TRAJ                    VAL WATSON                    2696
      SUBROUTINE TRAJ
C
C
C
C*************************************************************************
C
C           THIS SUBROUTINE CALCULATES THE CHANGE IN PARTICLE VELOCITIES AND
C                POSITIONS FROM THE EQUATIONS OF MOTION USING THE CURRENT
C                VALUES OF ELECTRIC POTENTIAL AND MAGNETIC FIELD
C
C*************************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(1800), RE(1800), ZI(1800), ZE(1800), VRI(1800), VRE(1800),
     1     VZI(1800), VZE(1800)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(6,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZESO,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)

      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1      NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST
      INTEGER INDXRC
      REAL RIO,ZIO,VRIO,VZIO,ERL,EZL,BL,ETA,THETA,ALPHA,THETA2,
     1   C1,C2,C3,C4
C *************************************************************
      DATA INDXRC/4/
      DATA ATHRST/0.0/
      DATA AEXITF/0.0/
      VRAND = SQRT(ABS(2.0*Q*VOLTS/MASSI)*0.1)
      THRUST = 0.0
      EXITFL = 0.0
      NRECOM = 0
      IF(NIT.EQ.0) GO TO 300
      FZA = ZA
      ZANS = FZA - 1.5
      ZANE = FZA - 0.5
C
C
C *** ION TRAJECTORIES ***
C
C
      ETA = DT*Q/MASSI
      DO 100 N=1,NIT
C
C
```

```
C *** CHANGING FROM ARRAYS TO INDIVIDUAL LOCATIONS ***
C
C
      RIO = RI(N)
      ZIO = ZI(N)
      VRIO = VRI(N)
      VZIO = VZI(N)
C
C
C *** CALCULATION OF LOCAL FIELDS ***
C
C
      I = RIO + 1.0
      J = ZIO + 1.0
      FI = I-1
      FJ = J-1
      IF(I.EQ.5) GO TO 2
      IF(J.EQ.15) GO TO 4
      ERL = ER(I,J) +(RIO - FI) * (ER(I+1,J) - ER(I,J))
     1   + (ZIO - FJ) * (ER(I,J+1) - ER(I,J))
      EZL = EZ(I,J) + (RIO - FI) * (EZ(I+1,J) - EZ(I,J))
     1     + (ZIO - FJ) *(EZ(I,J+1) - EZ(I,J))
      BL = B(I,J) + (RIO - FI) * (B(I+1,J) - B(I,J))
     1    + (ZIO - FJ) * (B(I,J+1) - B(I,J))
      GO TO 6
    2 CONTINUE
      ERL = ER(I,J) + (ZIO - FJ) * (ER(I,J+1) - ER(I,J))
      EZL = EZ(I,J) + (ZIO - FJ) * (EZ(I,J+1) - EZ(I,J))
       BL =  B(I,J) + (ZIO - FJ) * ( B(I,J+1) -  B(I,J))
      GO TO 6
    4 CONTINUE
      ERL = ER(I,J) + (RIO - FI) * (ER(I+1,J) - ER(I,J))
      EZL = EZ(I,J) + (RIO - FI) * (EZ(I+1,J) - EZ(I,J))
       BL =  B(I,J) + (RIO - FI) * ( B(I+1,J) -  B(I,J))
    6 CONTINUE
```

```
C
C
C *** CALCULATION OF THE CONSTANTS ***
C
C
      THETA = ETA*BL/2.0
      THETA2 = THETA**2
      C3 = 1.0/(1.0+THETA2)
      C1 = (1.0-THETA2)*C3
      C4 = THETA*C3
      C2 = 2.0*C4
C
C
C *** TRAJECTORY EQUATIONS ***
C
C
      VZIN= C1*VZIO + C2*VRIO + C3*ETA*EZL + C4*ETA*ERL
      VRIN = -C2*VZIO + C1*VRIO - C4*ETA*EZL + C3*ETA*ERL
      RIN = RIO + DT*VRIN/DR
      ZIN = ZIO + DT*VZIN/DZ
C
C
C *** BOUNDARY CONDITIONS FOR THE PARTICLES ***
C
C
      IF(RIN.LT.4.0) GO TO 10
           RIN = 8.0  - RIN
           VRIN = -REFLC*VRIN
      VZIN = REFLC * VZIN
      GO TO 20
   10 CONTINUE
      IF(RIN.GT.0.5) GO TO 20
      IF(ZIN.GT.7.0) GO TO 18
      INDXRC = INDXRC - 1
      IF(INDXRC.GE.0) GO TO 16
      INDXRC = 4
      NLI = NLI + 1

      NDELI(NLI) = N
      NRECOM = NRECOM + 1
      GO TO 30
   16       RIN = 1.0 - RIN
            VRIN = -VRIN * REFLC
      GO TO 20
   18 IF(RIN.GT.0.0) GO TO 20
      RIN = -RIN
      VRIN = -VRIN
   20 CONTINUE
      IF(ZIN.LT.14.5) GO TO 30
            NLI = NLI + 1
            NDELI(NLI) = N
      THRUST = THRUST + VZI(N)
      EXITFL = EXITFL + 1.0
      EXTRA1 = EXTRA1 + Q * QAMPL
      EXTRA1 = EXTRA1 - Q * QAMPL
      GO TO 40
   30 CONTINUE
      IF(ZIN.GT.0.0) GO TO 40
            ZIN = -ZIN
            VZIN = -REFLC*VZIN
      VRIN = REFLC * VRIN
   40 CONTINUE
C
C
C *** CHANGING BACK TO ARRAYS ***
C
C
      RI(N) = RIN
      ZI(N) = ZIN
      VRI(N) = VRIN
      VZI(N) = VZIN
  100 CONTINUE
```

```
C
C
C *** ELECTRON TRAJ¯CTORIES ***
C
C
      ETA =-DT*Q/MASSE
      DO 200 N=1,NET
C
C
C *** CHANGING FROM ARRAYS TO INDIVIDUAL LOCATIONS ***
C
C
      RIO = RE(N)
      ZIO = ZE(N)
      VRIO = VRE(N)
      VZIO = VZE(N)
C
C
C *** CALCULATION OF LOCAL FIELDS ***
C
C
      I = RIO + 1.0
      J = ZIO + 1.0
      FI = I-1
      FJ = J-1
      IF(I.EQ.5) GO TO 102
      IF(J.EQ.15) GO TO 104
      ERL = ER(I,J) +(RIO - FI) * (ER(I+1,J) - ER(I,J))
     1   + (ZIO - FJ) * (ER(I,J+1) - ER(I,J))
      EZL = EZ(I,J) + (RIO - FI) * (EZ(I+1,J) - EZ(I,J))
     1      + (ZIO - FJ) *(EZ(I,J+1) - EZ(I,J))
      BL = B(I,J) + (RIO - FI) * (B(I+1,J) - B(I,J))
     1    + (ZIO - FJ) * (B(I,J+1) - B(I,J))
      GO TO 106
  102 CONTINUE
      ERL = ER(I,J) + (ZIO - FJ) * (ER(I,J+1) - ER(I,J))
      EZL = EZ(I,J) + (ZIO - FJ) * (EZ(I,J+1) - EZ(I,J))

       BL =  B(I,J) + (ZIO - FJ) * ( B(I,J+1) -  B(I,J))
      GO TO 106
  104 CONTINUE
      ERL = ER(I,J) + (RIO - FI) * (ER(I+1,J) - ER(I,J))
      EZL = EZ(I,J) + (RIO - FI) * (EZ(I+1,J; - EZ(I,J))
       BL =  B(I,J) + (RIO - FI) * ( B(I+1,J) -  B(I,J))
  106 CONTINUE
C
C
C *** CALCULATION OF THE CONSTANTS ***
C
C
      THETA = ETA*BL/2.0
      THETA2 = THETA**2
      C3 = 1.0/(1.0+THETA2)
      C1 = (1.0-THETA2)*C3
      C4 = THETA*C3
      C2 = 2.0*C4
C
C
C *** TRAJECTORY EQUATIONS ***
C
C
      VZIN= C1*VZIO + C2*VRIO + C3*ETA*EZL + C4*ETA*ERL
     1             + VRAND*(RANDOM(+1)-0.5)
      VRIN = -C2*VZIO + C1*VRIO - C4*ETA*EZL + C3*ETA*ERL
     1             + VRAND*(RANDOM(+1)-0.5)
      RIN = RIO + DT*VRIN/DR
      ZIN = ZIO + DT*VZIN/DZ
C
C
C *** BOUNDARY CONDITIONS FOR THE PARTICLES ***
C
C
```

```
      IF(                   ZIN.GT.7.00.OR.RIN.LT.4.0) GO TO 108

      QNET = QNET - NAMPL*Q
      NLE = NLE + 1
      NDELE(NLE) = N
      GO TO 140
  108 IF(RIN.LT.4.0) GO TO 110
          RIN = 8.0  - RIN
          VRIN = -REFLC*VRIN
      VZIN = REFLC * VZIN
      GO TO 120
  110 CONTINUE
      IF(ZIN.GT.7.0.OR.RIN.GT.0.5) GO TO 112
          RIN = 1.0 - RIN
          VRIN = -VRIN * REFLC
      GO TO 12C
  112 CONTINUE
      IF(ZIN.LE.7.0.OR.RIN.GT.0.0) GO TO 120
          RIN = -RIN
      VRIN = -VRIN
  120 CONTINUE
      IF(ZIN.LT.14.5) GO TO 130
          NLE = NLE + 1
          NDELE(NLE) = N
      GO TO 140
  130 CONTINUE
      IF(ZIN.GT.0.0) GO TO 140
          ZIN = -ZIN
          VZIN = -REFLC*VZIN
      VRIN = REFLC * VRIN
  140 CONTINUE
C
C
C *** CHANGING BACK TO ARRAYS ***
C
C
```

```
      RE(N) = RIN
      ZE(N) = ZIN
      VRE(N) = VRIN
      VZE(N) = VZIN
  200 CONTINUE
  300 CONTINUE
      THRUST = THRUST * NAMPL           * MASSI/DT
      ATHRST = (THRUST + 99. * ATHRST)/100.0
      EXITFL = EXITFL * NAMPL           * MASSI/DT
      AEXITF=(EXITFL + 99. * AEXITF)/100.0
      TTHRST = (AMPSI        *MASSI/Q)*SQRT(2.0*Q*VOLTS/MASSI)
      PTHRST =(THRUST/TTHRST) * 100.0
      PATHRS =(ATHRST/TTHRST) * 100.0
      TEXITF = AMPSI        *MASSI/Q
      PEXITF =(EXITFL/TEXITF) * 100.0
      PAEXIT =(AEXITF/TEXITF) * 100.0
      IF(OUTIND.GT.0) GO TO 500
      WRITE(6,400) TTHRST,PTHRST,PATHRS,TEXITF,PEXITF,PAEXIT
  400 FORMAT(1H1, 24HTHEORETICAL THRUST =       ,1PE15.4,10H NEWTONS    /
     1            24H ACTUAL THRUST =           ,2PE15.4,10H  PERCENT   /
     2            24H AVERAGE THRUST =          ,2PE15.4,10H  PERCENT   /
     3            25H THEORETICAL MASS FLOW =   ,1PE15.4,10H KG/SEC     /
     4            24H ACTUAL MASS FLOW =        ,2PE15.4,10H  PERCENT   /
     5            24H AVERAGE MASS FLOW =       ,2PE15.4,10H  PERCENT   )
  500 CONTINUE
      LAST = 1
      RETURN
      END
```

```
CMP04D5 SUBROUTINE SOURCE                      VAL WATSON                                    2696
      SUBROUTINE SOURCE
C
C
C
C***********************************************************************
C
C        THIS SUBROUTINE PROVIDES FOR THE INJECTION OF PARTICLES INTO THE
C             ACCELERATOR BY VOLUME IONIZATION OR FROM THE ACCELERATOR
C             BOUNDARIES
C
C***********************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(1800), RE(1800), ZI(1800), ZE(1800), VRI(1800), VRE(1800),
     1     VZI(1800), VZE(1800)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(6,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZE$O,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)

      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1     NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST,NLOC,NSINDI,NSINDE
      REAL RSRCE(100),ZSRCE(100)
C ************************************************************
C
C
C *** ION SOURCE ***
      DATA NSINDI,NSINDE/1,1/
      DATA TLADD,TLADP,TLADDE,TLADPE/0.0,0.0,0.0,0.0/
      DATA FIRST/0/
      IF(FIRST.GT.0) GO TO 1
      FIRST = 1
C
C *** GENERATE THE ARBITRARY SOURCE FOR IONIZATION ***
C
      NLOC = 99
      DO 1000 I = 1,99
       RSRCE(I) = RANDOM(+1) * 3.5 + 0.5
      ZSRCE(I) = - ALOG(1.0-RANDOM(+1) + EXP(-14.0))
 1000 CONTINUE
      TLADP = (AMPSI*(T-DT))/(NAMPL*Q)
      TLADPE= (AMPSE*(T-DT))/(NAMPL*Q)
    1 CONTINUE
  300 FORMAT(I4/(2E10.3))
C *** DIFFERENCE BETWEEN THE TOTAL LOST AND THE TOTAL INPUT ***
      TLADD = TLADD + (AMPSI * DT)/(NAMPL * Q)
      NII = IFIX(TLADD) - IFIX(TLADP) + NRECOM

      TLADP = TLADD
C
C
      NDI = NII - NLI
C
C
C *** ASSIGNING PARTICLES TO THE N NUMBERS ***
C
C
C
C
C *** ASSIGNING NEW PARTICLES OF EACH RING ***
C
C
      IF(NII.LE.0) GO TO 20
      DO 6 NIND=1,NII
      IF(NLI.GT.0) GO TO 2
C
C *** SPACES NOT LEFT IN OLD ARRAY ***
C
C
      NIT = NIT+1
      IF(NIT.GT.1800) GO TO 400
      N = NIT
      GO TO 4
    2 CONTINUE
C
C
C *** SPACES LEFT IN OLD ARRAY ***
C
C
      N = NDELI(NLI)
      NLI = NLI-1
    4 CONTINUE
```

```
      RI(N) = RSRCE(NSINDI)

      ZI(N) = ZSRCE(NSINDI)
      NSINDI = NSINDI + 1
      IF(NSINDI.GT.NLOC) NSINDI = 1
      VRI(N)=(RANDOM(+1)-0.5)*(SQRT(2.0*Q*VOLTS/MASSI)*0.1)
      VZI(N)=(RANDOM(+1)-0.5)*(SQRT(2.0*Q*VOLTS/MASSI)*0.1)
    6 CONTINUE
   20 IF(NLI.LE.0) GO TO 30
C
C
C *** IF SPACES REMAIN AFTER ASSIGNING ALL NEW PARTICLES ***
C
C
      N = NDELI(NLI)
      NLI = NLI-1
      RI(N) = RI(NIT)
      ZI(N) = ZI(NIT)
      VRI(N) = VRI(NIT)
      VZI(N) = VZI(NIT)
      NIT = NIT-1
      GO TO 20
   30 CONTINUE
C
C
C *** ELECTRON SOURCE ***
      TLADDE = TLADDE + (AMPSE * DT)/(NAMPL * Q)
      NIE = IFIX(TLADDE) - IFIX(TLADPE)
      TLADPE = TLADDE
      NIE = NIE + NII - NRECOM
      FNREC = NRECOM
      QNET = QNET + NAMPL*Q*FNREC
C *** DIFFERENCE BETWEEN THE TOTAL LOST AND THE TOTAL INPUT ***
C
C
      NDE = NIE - NLE
```

```
C

C
C *** ASSIGNING PARTICLES TO THE N NUMBERS ***
C
C
C
C
C *** ASSIGNING NEW PARTICLES OF EACH RING ***
C
C
      IF(NIE.LE.0) GO TO 120
      DO 106 NIND = 1,NIE
      IF(NLE.GT.0) GO TO 102
C
C *** SPACES NOT LEFT IN OLD ARRAY ***
C
C
      NET = NET+1
      IF(NET.GT.1800) GO TO 400
      N = NET
      GO TO 104
  102 CONTINUE
C
C
C *** SPACES LEFT IN OLD ARRAY ***
C
C
      N = NDELE(NLE)
      NLE = NLE-1
  104 CONTINUE
      IF(NIND.GT.NII) GO TO 105
      RE(N) = RSRCE(NSINDE)
      ZE(N) = ZSRCE(NSINDE)
      NSINDE = NSINDE + 1
      IF(NSINDE.GT.NLOC) NSINDE = 1
      VRE(N)=(RANDOM(+1)-0.5)*(SQRT(2.0*Q*VOLTS/MASSI)*0.1)

      VZE(N)=(RANDOM(+1)-0.5)*(SQRT(2.0*Q*VOLTS/MASSI)*0.1)
      GO TO 106
  105 RE(N) = 0.501
      ZE(N) = RANDOM(+1) * 7.0
      VRE(N)=(RANDOM(+1)     )*(SQRT(2.0*Q*VOLTS/MASSI)*0.1)
      VZE(N)=(RANDOM(+1)-0.5)*(SQRT(2.0*Q*VOLTS/MASSI)*0.1)
      QNET = QNET + NAMPL*Q
  106 CONTINUE
  120 IF(NLE.LE.0) GO TO 130
C
C
C *** IF SPACES REMAIN AFTER ASSIGNING ALL NEW PARTICLES ***
C
C
      N = NDELE(NLE)
      NLE = NLE-1
      RE(N) = RE(NET)
      ZE(N) = ZE(NET)
      VRE(N) = VRE(NET)
      VZE(N) = VZE(NET)
      NET = NET-1
      GO TO 120
  130 CONTINUE
      GO TO 500
  400 NEXCD = 1
  500 CONTINUE
  302 FORMAT(1H0,20H SOURCE LOCATIONS   /(1P2E10.3))
      LAST = 1
      RETURN
      END
```

```
CMP04D6 SUBROUTINE OUTPT                  VAL WATSON                         2696
      SUBROUTINE OUTPT
C
C
C
C**********************************************************************
C
C         THIS SUBROUTINE PRINTS THE LOCAL VALUES OF THE FIELDS,
C              DENSITIES, AND CURRENTS
C
C**********************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(1800), RE(1800), ZI(1800), ZE(1800), VRI(1800), VRE(1800),
     1     VZI(1800), VZE(1800)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(6,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZESO,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)
      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1      NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST
C ***************************************************************
C
      IF(TSTEP.EQ.0.AND.T.GT.0.0) GO TO 200
      NFILE = NFILE + 1
      WRITE(8)  T,PHI,B,NUMI,NUME,NUMVI,NUMVE,NIT,(RI(I),ZI(I),VRI(I),VZ
     1I(I),I=1,NIT),NET,(RE(I),ZE(I),VRE(I),VZE(I),I=1,NET),NFRAME,QNET
      END FILE 8
  200 WRITE(6,300) T,NFILE,NFRAME
      WRITE(6,310)
      WRITE(6,320) ((PHI(I,J), I=1,5),J=1,15)
      WRITE(6,330)
      WRITE(6,320) ((    B(I,J),I=1,5),J=1,15)
      WRITE(6,340)
      WRITE(6,320) ((NUMI(I,J),I=1,5),J=1,15)
      WRITE(6,350)
      WRITE(6,320) ((NUME(I,J),I=1,5),J=1,15)
      WRITE(6,360)
      WRITE(6,320) ((NUMVI(I,J),I=1,5),J=1,15)
      WRITE(6,370)
      WRITE(6,320) ((NUMVE(I,J),I=1,5),J=1,15)
  300 FORMAT(1H1,  10H TIME =    , 1PE15.5, 5X, 8H FILE NO  ,I4,
     1      5X, 9H FRAME NO   , I4)
  310 FORMAT(1H0, 20H *** POTENTIALS ***              )
  320 FORMAT(1P5E13.5)
  330 FORMAT(1H0, 30H *** MAGNETIC FIELDS ***                  )
  340 FORMAT(1H1, 40H *** NUMBER OF IONS IN EACH CELL ***          )
```

```
350 FORMAT(1H0, 40H *** NUMBER OF ELECTRONS PER CELL ***         )

360 FORMAT(1H1, 40H *** MASS FLOW OF IONS ***                    )
370 FORMAT(1H0, 40H *** MASS FLOW OF ELECTRONS ***               )
    LAST = 1
    RETURN
    END
```

```
CMP04D7 SUBROUTINE OTTPT                VAL WATSON                        2696
      SUBROUTINE OTTPT
C
C
C
C**********************************************************************
C
C          THIS SUBROUTINE CONVERTS THE VALUES FOR THE PARTICLE POSITIONS
C               AND THE FIELDS TO 12 BIT INTEGERS AND STORES 3 PER WORD
C               ON MAGNETIC TAPE FOR FUTURE PLOTTING
C
C**********************************************************************
C
C
C
      COMMON/AA/RI,RE,ZI,ZE,VRI,VRE,VZI,VZE
      COMMON/BB/RAD,VOL,NUME,NUMI,NUMVE,NUMVI,RHO,JZ,PHI,A,B,EZ,ER
      COMMON/CC/T,DT,MASSR,NAMPL,EPSAMP,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZERO
     1 ,EPSO,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      COMMON/DD/NSE,NSI,NDELI,NDELE
      COMMON/EE/TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     2   NLI,NLF,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD,RA,RC,ZA,ZC
      COMMON/FF/EXTRA1,EXTRA2,EXTRA3,NXTRA1,NXTRA2,NXTRA3
C
C
C *** COMMON VARIABLES ***
C
C
      REAL RI(1800), RE(1800), ZI(1800), ZE(1800), VRI(1800), VRE(1800),
     1     VZI(1800), VZE(1800)
      REAL RAD(5), VOL(5), NUME(5,15), NUMI(5,15), NUMVE(5,15),
     1     NUMVI(5,15), RHO(6,15), JZ(5,15), PHI(5,15), A(5,15), B(5,15)
     1     ,EZ(5,15), ER(5,15)
      REAL T,DT,MASSR,NAMPL,EPSAMF,REFLC,VOLTS,EX,DR,DZ,PI,Q,MUZESO,EPSO
     1     ,MASSI,MASSE,RDS,LENG,AMPSE,AMPSI,QNET
      REAL EXTRA1,EXTRA2,EXTRA3
      INTEGER NSE(5,15), NSI(5,15), NDELI(400), NDELE(400)

      INTEGER TSTEP,NTSTEP,OUTINC,OTTINC,OUTIND,OTTIND,NFILE,NIT,NET,
     1      NLI,NLE,NII,NIE,N,NDI,NDE,NFRAME,NRECOM,NEXCD
      INTEGER RA,RC,ZA,ZC
      INTEGER NXTRA1,NXTRA2,NXTRA3
C
C
C *** LOCAL VARIABLES ***
C
C
      INTEGER FIRST
      INTEGER IARRAY(770)
C **************************************************************
C
      IF(TSTEP.GT.1) GO TO 100
      DO 60 IA = 1,380
      IARRAY(IA) = 0
   60 CONTINUE
      READ(5,500) TAMP,PHIAMP,BAMP,RAMP,ZAMP
      WRITE(6,540) TAMP,PHIAMP,BAMP,RAMP,ZAMP
  100 CONTINUE
      IF(TSTEP.EQ.1) GO TO 600
      IARRAY(1) = 0
      IARRAY(2) = T*TAMP
      IARRAY(3) = NIT
      IARRAY(4) = NET
      IA = 4
C
C *** CONVERT INFORMATION TO BE PLOTTED INTO 12 BIT INTEGERS ***
C
      DO 140 J = 1,15
      DO 120 I = 1,5
            IA = IA + 1
            IARRAY(IA) = PHI(I,J)*PHIAMP + 200.0
  120 CONTINUE
  140 CONTINUE
      DO 200 J = 1,15
      DO 180 I = 1,5
            IA = IA + 1
            IARRAY(IA) = B(I,J)*BAMP + 200.0
  180 CONTINUE
  200 CONTINUE
      CALL   FOLD(IARRAY(2),153)
C
C        SUBROUTINE FOLD PACKS 3 12 BIT INTEGERS INTO EACH 7094 BIT WORD ***
C
      WRITE(10) (IARRAY(I),I=1,53)
      DO 240 I=1,770
  240  IARRAY(I) = 0
      IA = 1
      DO 300 I=1,NIT
            IA = IA + 1
            IARRAY(IA) = RI(I)*RAMP
            IA = IA + 1
            IARRAY(IA) = ZI(I)*ZAMP
            IF(IA.LT.763) GO TO 300
            CALL   FOLD(IARRAY(2),762)
            WRITE(10) (IARRAY(J),J=1,255)
            IA = 1
  300 CONTINUE
      IF(IA.EQ.1) GO TO 320
      IARL = 1000 - (3000-IA+1)/3
      IARL3 = IARL*3
      IARLP = IARL+1
      IA = IA+1
      IARRAY(IA) = 0
      IA = IA+1
      IARRAY(IA) = 0
      CALL FOLD(IARRAY(2),IARL3)
      WRITE(10) (IARRAY(I),I=1,IARLP)
  320 CONTINUE
```

```
      IA = 1

      DO 400 I=1,NET
             IA = IA + 1
             IARRAY(IA) = RE(I)*RAMP
             IA = IA + 1
             IARRAY(IA) = ZE(I)*ZAMP
             IF(IA.LT.763) GO TO 400
             CALL   FOLD(IARRAY(2),762)
             WRITE(10) (IARRAY(J),J=1,255)
             IA = 1
  400 CONTINUE
      IF(IA.EQ.1) GO TO 420
      IARL = 1000 - (3000-IA+1)/3
      IARL3 = IARL*3
      IARLP = IARL+1
      IA = IA+1
      IARRAY(IA) = 0
      IA = IA+1
      IARRAY(IA) = 0
      CALL FOLD(IARRAY(2),IARL3)
      WRITE(10) (IARRAY(I),I=1,IARLP)
  420 CONTINUE
      NFRAME = NFRAME + 1
      LAST = 1
  500 FORMAT(5E10.3)
  540 FORMAT(2H , 9HTAMP    =    ,E10.3/ 10H PHIAMP =    ,E10.3/
     1          10H BAMP    =    ,E10.3/ 10H RAMP   =    ,E10.3/
     2          10H ZAMP    =    ,E10.3)
  600 RETURN
      END
```

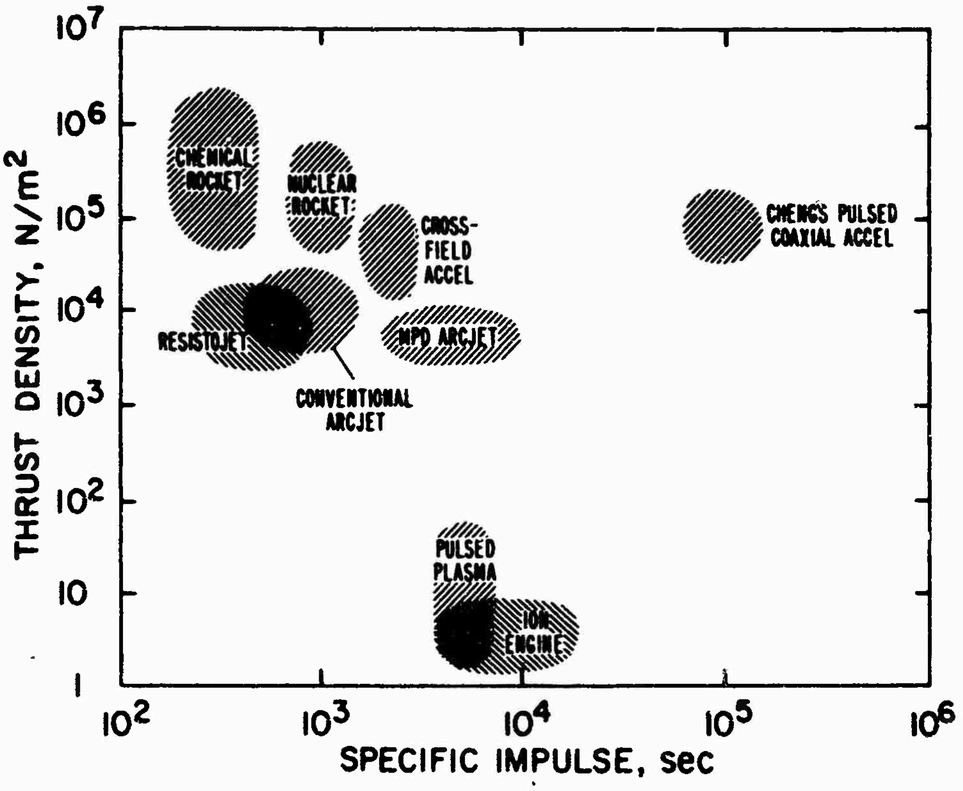


Figure 1. Performance of Various Propulsion Devices as Depicted by Clark and Jahn[1]

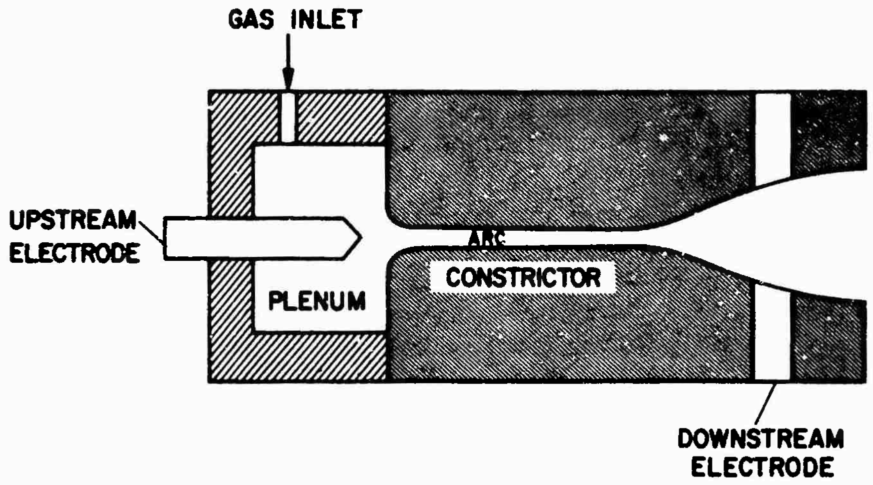


Figure 2. A Conventional Constricted Arcjet

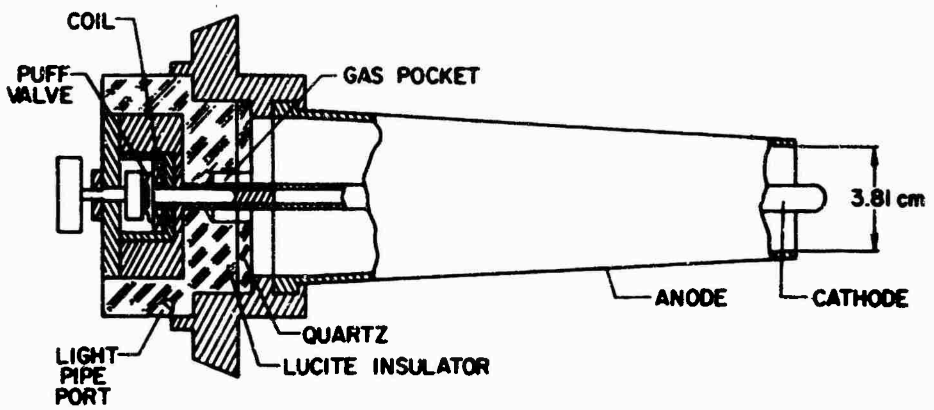


Figure 3. Cheng's Pulsed Coaxial Accelerator[2]

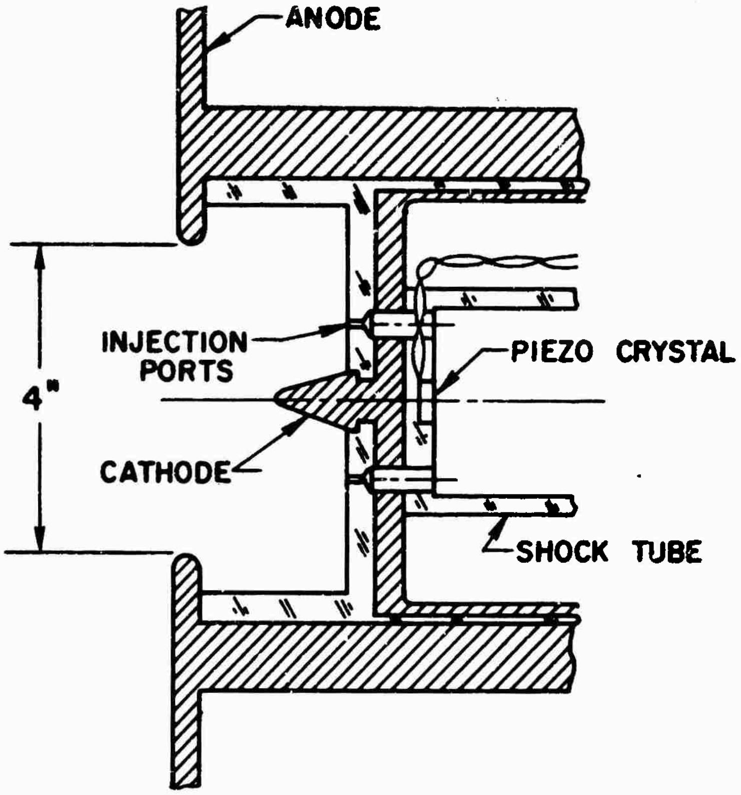


Figure 4. Cross Section of Clark and Jahn's Pulsed Accelerator

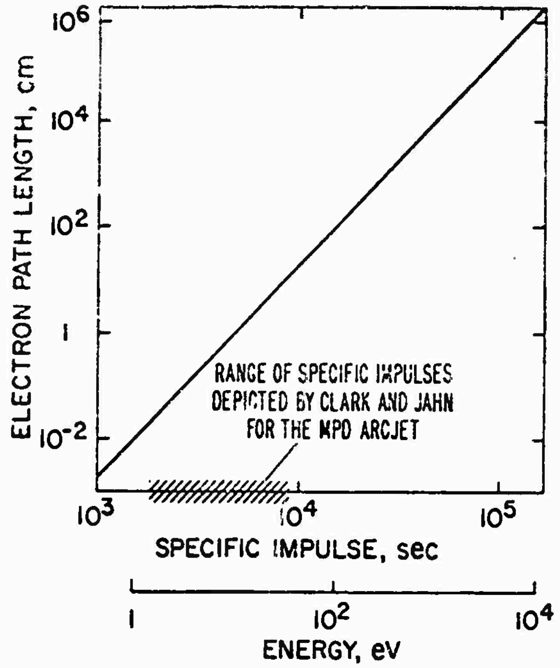


Figure 5. Electron Path Length Required to Transfer Energy to Hydrogen Ions with Ion Number Densities of 10^{15}ions/cc

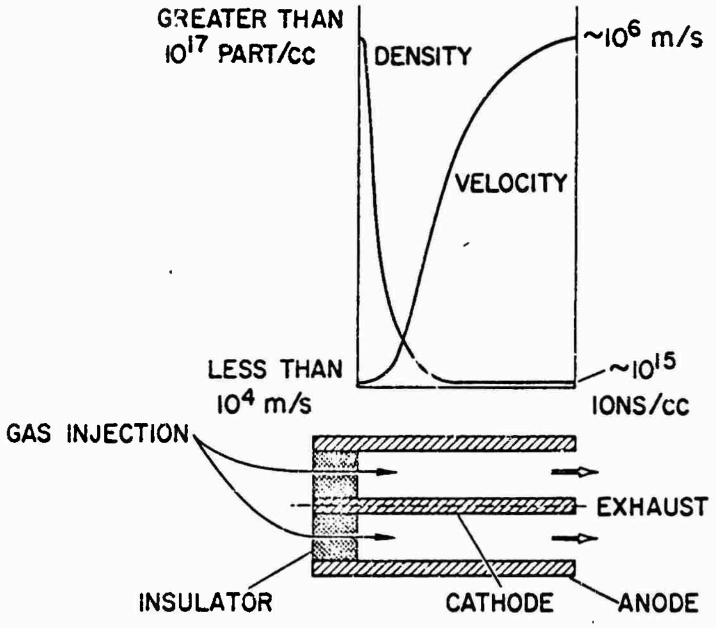


Figure 6. Hypothetical Density and Velocity Distributions Within Cheng's Coaxial Accelerator

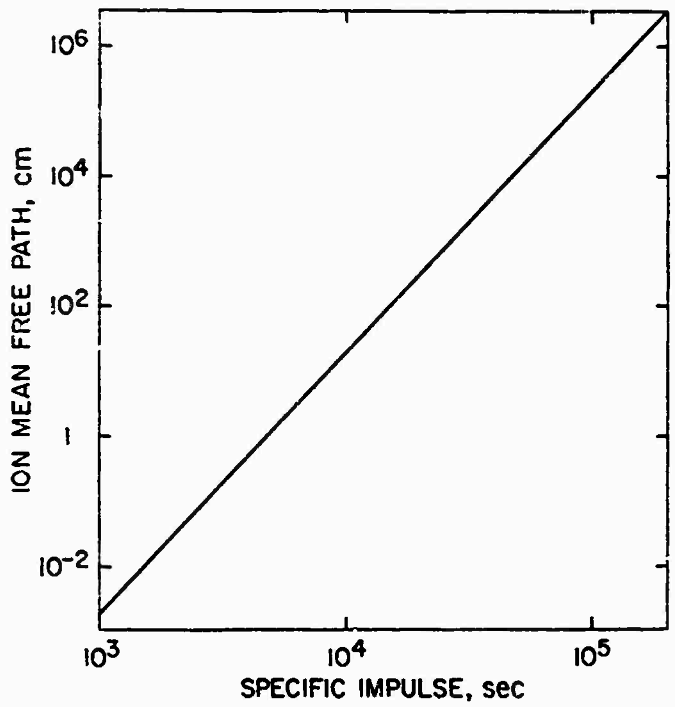


Figure 7. Mean Free Paths of Hydrogen Ions at Ion Number Densities of 10^{15} ions/cc

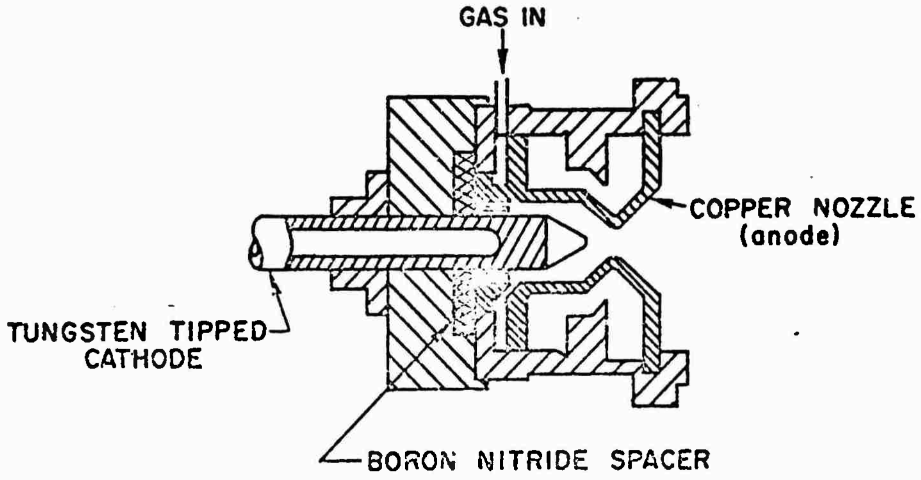


Figure 8. A Typical MPD Accelerator (From Jahn[11])

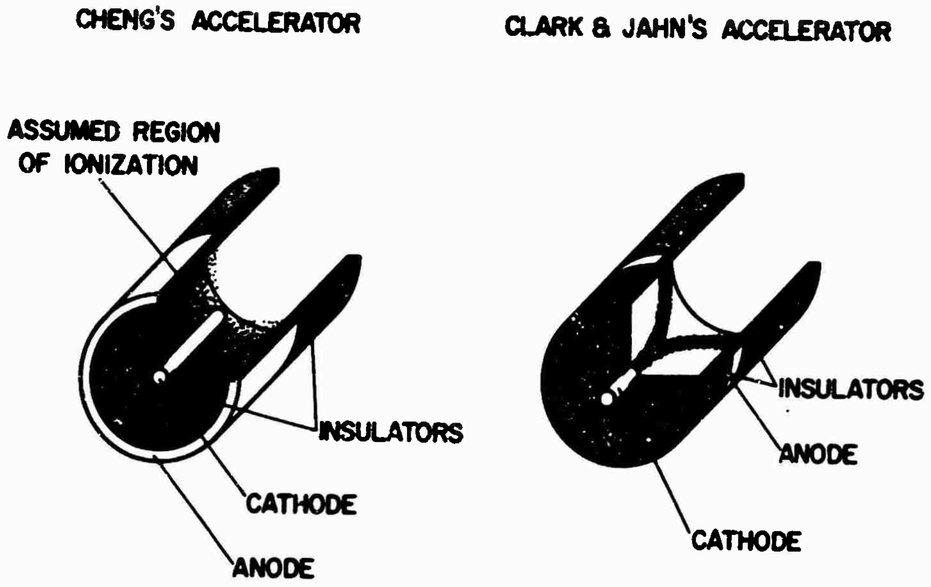


Figure 9. Accelerator Models Used in the Computer Simulations

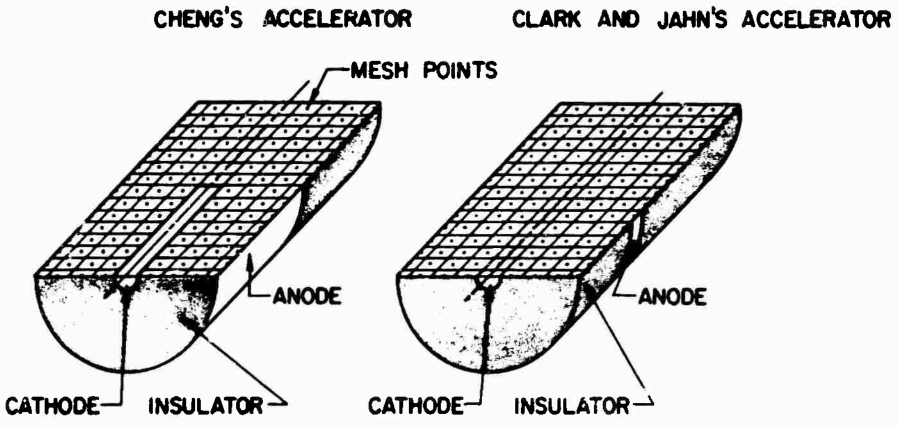
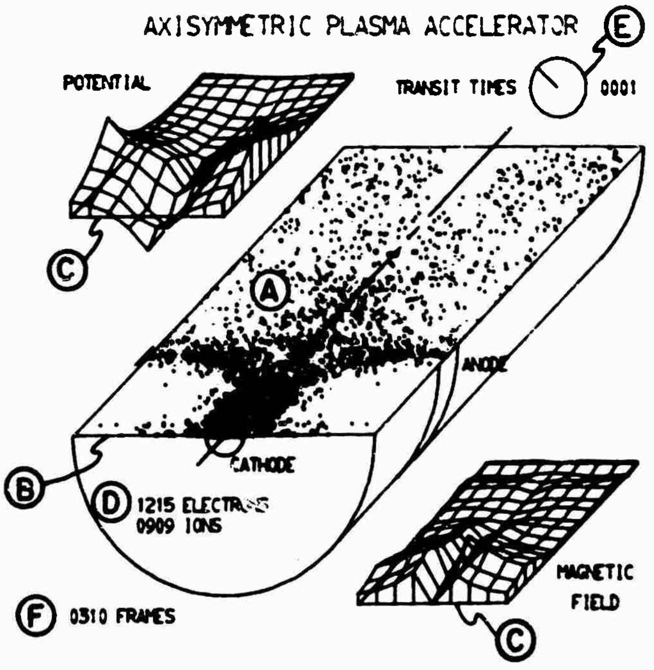


Figure 10. Mesh Configuration Within the Models of the Accelerators Used for the Computer Simulation



LEGEND

A. The representative particles simulating the plasma; the large dots represent ions and the small dots represent electrons.

B. The reference plane of constant aximuthal position showing the radial and axial positions within the accelerator.

C. The same reference plane translated to a new position in order to plot the electric potential and the magnetic field as a function of radial and axial position. For each point in the reference plane, the vertical height to the surface plotted represents the magnitude of the electric potential or of the magnetic field for that radial and axial position.

D. The number of representative electrons and ions currently being used to simulate the plasma.

E. A clock measuring the time in units of transit times. A transit time is the time that would be required for an ion to travel the length of the accelerator if it had an energy in electron volts equal to the potential difference between the electrodes in volts.

F. The number of movie frames from the start of each case.

Figure 11. Sample Frame of the Movie of the Plasma Simulation

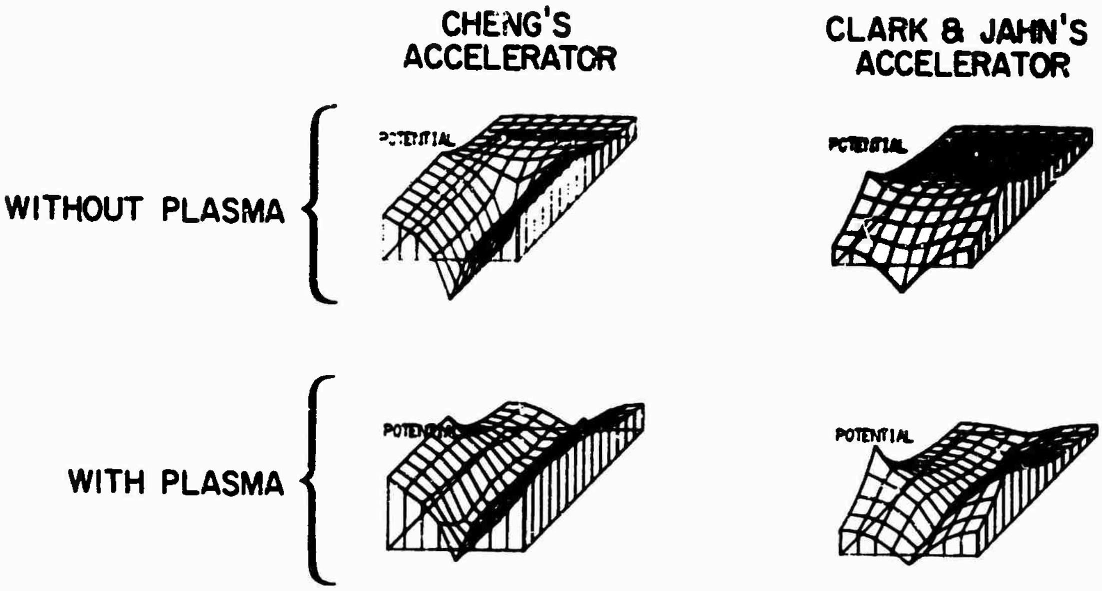


Figure 12. Modification of the Applied Potentials by the Plasma

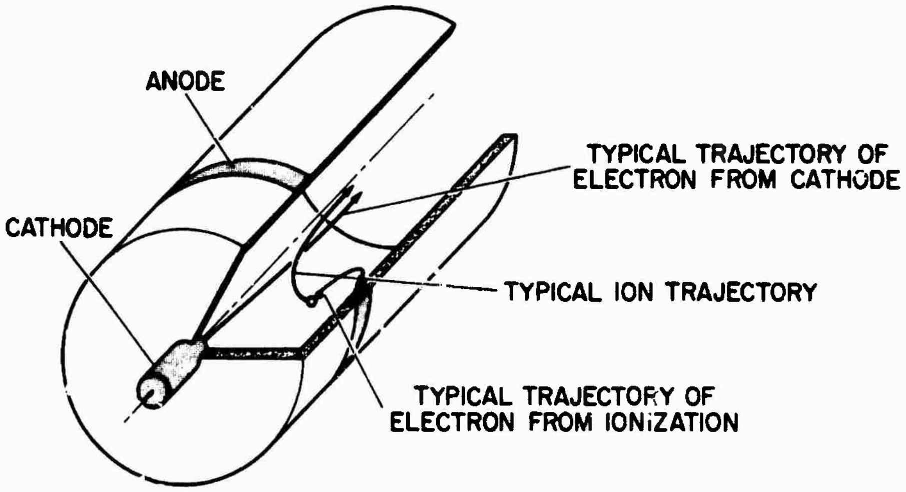

ANODE
CATHODE
TYPICAL TRAJECTORY OF ELECTRON FROM CATHODE
TYPICAL ION TRAJECTORY
TYPICAL TRAJECTORY OF ELECTRON FROM IONIZATION

Figure 13. Typical Particle Trajectories Observed in the Computer Simulation.

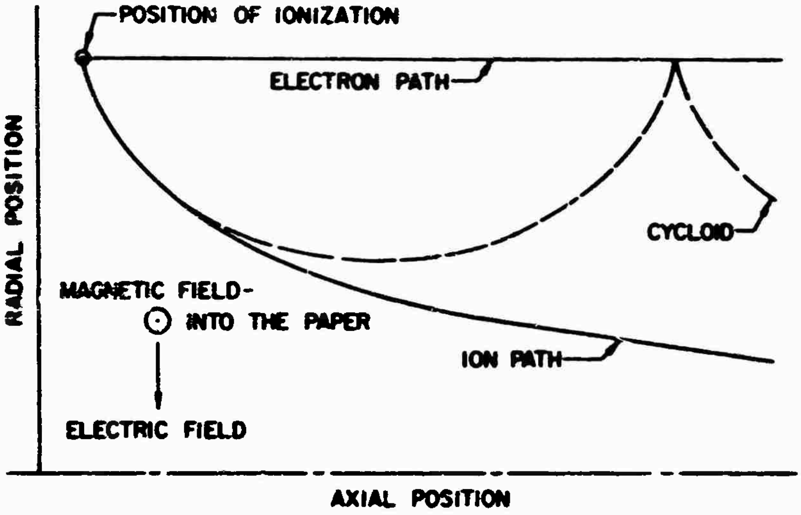


Figure 14. Electron and Ion Trajectories in a Uniform Electric Field And A Magnetic Field That Decays in the Axial Direction

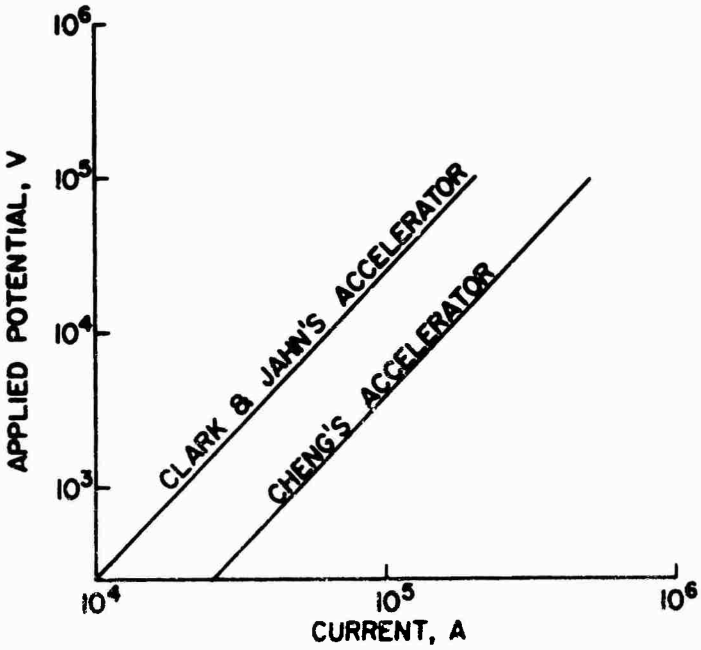


Figure 15. Voltage-Current Characteristic, Obtained from the Computer Simulation

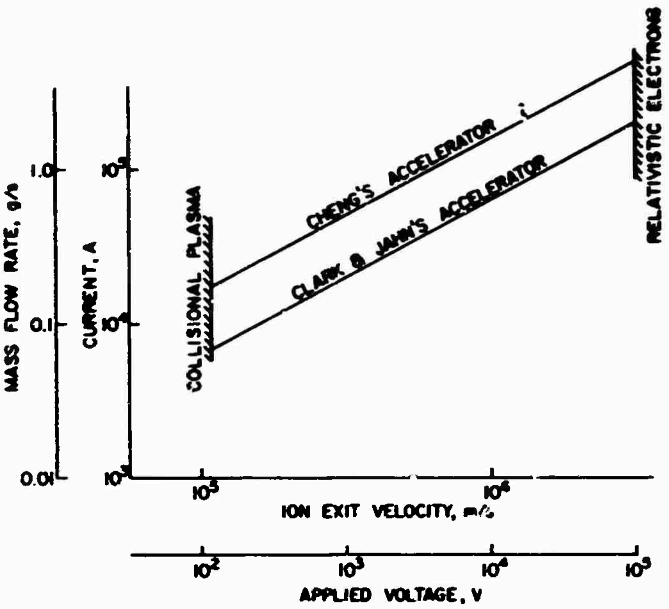


Figure 16. Performance Predictions Obtained from the Computer Simulations

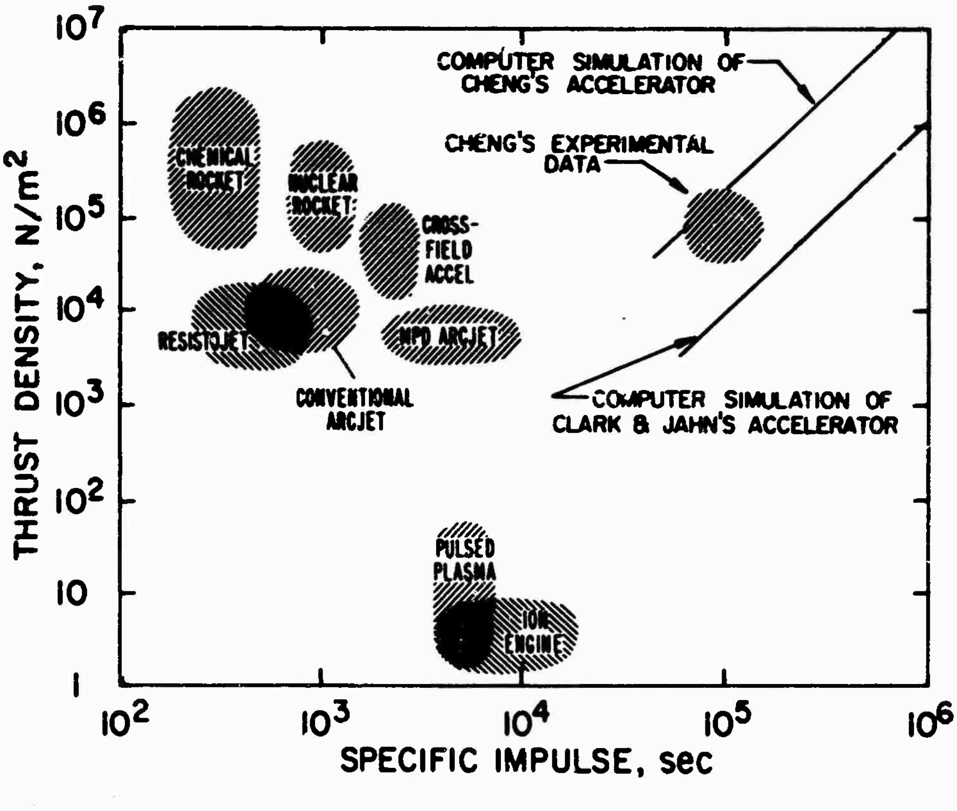


Figure 17. Performance Predictions Obtained from the Computer Simulations Compared with Cheng's Experimental Data and With the Performance of Various Propulsion Devices as Depicted by Clark and Jahn[1]